TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

RASMUS MATTILA

MODEL PREDICTIVE CONTROL WITH FIXED SWITCHING FREQUENCY FOR AC MOTOR DRIVES

Master of Science thesis

Examiner: Assist. Prof. Petros Karamanakos
Examiner and topic approved by the Faculty Council of the Faculty of Computing and Electrical Engineering on 30th November 2017

ABSTRACT

RASMUS MATTILA: Model Predictive Control with Fixed Switching Frequency for Ac Motor Drives
Tampere University of Technology
Master of Science thesis, 52 pages, 1 Appendix page
January 2018
Master's Degree Programme in Electrical Engineering
Major: Power Electronics
Examiner: Assist. Prof. Petros Karamanakos
Keywords: Model predictive control, Ac drive, Direct model predictive control, Fixed switching frequency

Model predictive control (MPC) is a control strategy that seems promising in the area of ac drives. In MPC, the model of the plant is used to predict its future behavior; the control action is determined based on the most desirable behavior of the system. MPC can be applied to multiple-input multiple-output (MIMO) systems and systems with switching nature, e.g. ac drives. Furthermore, limitations and restrictions can be implemented easily in the form of constraints.

This thesis presents an MPC algorithm for motor control that can operate the power electronic converter at fixed switching frequency. This is achieved despite the fact that the switches of the inverter are controlled directly. The performance of the proposed method is compared to other MPC algorithms as well as field oriented control (FOC) with carrier-based pulse width modulation (CB-PWM). The method is able to outperform CB-PWM during transients, whereas the steady-state behavior is similar. The other MPC algorithms outperform the proposed method in terms of stator current total harmonic distortion (THD), but unlike the proposed method and CB-PWM, they do not produce a constant switching frequency.

CONTENTS

LIST OF ABBREVIATIONS AND SYMBOLS

ac	alternating current
CB-PWM	Carried-Based Pulse Width Modulation
dc	direct current
FCS-MPC	Finite Control Set Model Predictive Control
FOC	Field Oriented Control
FPGA	Field Programmable Gate Array
IM	Induction Machine
LV	Low Voltage
MIMO	Multiple-Input Multiple-Output
MPC	Model Predictive Control
MV	Medium Voltage
NPC	Neutral Point Clamped
p.u.	per unit
PI	Proportional Integral
PWM	Pulse Width Modulation
QP	Quadratic Program
rms	root mean square
SISO	Single-Input Single-Output
THD	Total Harmonic Distortion
VSD	Variable Speed Drive
VSP^{2}CC	Variable Switching Point Predictive Current Control

$\mathbf{0}$	zero vector
$\boldsymbol{A}$	discrete-time state-space system matrix
$\boldsymbol{B}$	discrete-time state-space input matrix
$\boldsymbol{C}$	discrete-time/continuous-time state-space output matrix
D	determinant
e_{rms}	squared root mean square error
f_c	carrier frequency
f_f	fundamental frequency
f_{sw}	average switching frequency
$\boldsymbol{F}$	continuous-time state-space system matrix
$\boldsymbol{G}$	continuous-time state-space input matrix
$\boldsymbol{i}_s$	stator current in $\alpha\beta$
$\boldsymbol{i}_{s,abc}$	stator current in abc
$\boldsymbol{J}$	cost function

k	discrete-time time-step
$\boldsymbol{K}$	reduced Clarke transformation matrix
$\boldsymbol{K}^{-1}$	inverse reduced Clarke transformation matrix
$\boldsymbol{K}(\varphi)$	reduced Park transformation matrix
$\boldsymbol{K}^{-1}(\varphi)$	inverse reduced Park transformation matrix
m	modulation index
$\boldsymbol{m}$	vector of constant slopes in $\alpha\beta$
N_P	prediction horizon length
p	number of pole pairs
r_r	rotor resistance in p.u.
r_s	stator resistance in p.u.
R_r	rotor resistance
R_s	stator resistance
$\boldsymbol{R}(\varphi)$	rotational matrix
$\boldsymbol{R}^{-1}(\varphi)$	inverse rotational matrix
t	time
t_{sw}	time to switch
$\boldsymbol{t}_{\mathrm{sw}}$	vector of times to switch
T_c	carrier period
T_f	fundamental period
T_s	sampling interval
$\hat{u}^*$	peak value of the scaled voltage reference
$\boldsymbol{u}$	input vector, vector of three-phase switch positions
$\boldsymbol{u}^*$	scaled three-phase voltage reference
$\boldsymbol{U}$	sequence of control actions/inputs
x_{ls}	stator leakage reactance in p.u.
x_m	mutual leakage reactance in p.u.
x_{rs}	rotor leakage reactance in p.u.
$\boldsymbol{x}$	state vector
X_{ls}	stator leakage reactance
X_m	mutual leakage reactance
X_{rs}	rotor leakage reactance
X_σ	total leakage reactance
$\boldsymbol{y}$	output vector
$\hat{v}^*$	peak value of the stator voltage reference
$v_{sd,ff}$	stator voltage feed-forward term d component in FOC
$v_{sq,ff}$	stator voltage feed-forward term q component in FOC
$\boldsymbol{v}$	stator voltage in $\alpha\beta$
$\boldsymbol{v}^*$	three-phase voltage reference

V_{dc}	dc link voltage
λ	weighting factor
ω_{fr}	angular speed of the reference frame
ω_r	rotor side angular speed
ω_s	stator side angular speed
$\boldsymbol{\psi}_r$	rotor flux
τ	time constant
φ	angular position of a reference frame

1. INTRODUCTION

Model predictive control (MPC) is a method for controlling a system by using a model to predict its behavior. By doing so, the input that produces the best possible output can be chosen.

MPC has been used in many industries, e.g. process industry, for decades [22]. Compared to those applications, power electronic systems have significantly shorter time constants, which correlates to shorter sampling intervals. In MPC, the necessary computation has to be done within each sampling interval. Additionally, the switching nature of power electronic systems complicates their control. In other words, to use MPC for power electronic systems, a lot of computation needs to take place in a short amount of time.

Moore's observation [7] regarding the number of transistors in an area of a circuit has held up [4]. The computational power available has reached a point where MPC for power electronics is possible. Consequently, MPC for power electronics has been a topic of research [25].

MPC can be used with systems that are linear or non-linear and single-input single-output (SISO) or multiple-input multiple-output (MIMO). For example, a variable speed drive (VSD) is a MIMO system: it needs more than one input and output signals. Modern control strategies, such as FOC, use proportional integral (PI) controllers. As PI controllers are SISO controllers, cascaded control loops are utilized to control multiple interdependent variables. Cascaded control loops perform sub-optimally under transients, especially under operating conditions that greatly differ from the conditions that the controllers were tuned in. MPC can be used with MIMO systems with no need for cascaded control loops [8]. Additionally, MPC reduces the control design effort at the cost of more computational complexity [17]. At low switching frequencies, MPC outperforms traditional control methods even in steady-state conditions [11].

The motor side control of a VSD requires the switches of an inverter to be turned on and off. If the control is done indirectly, this is done based on a modulation

scheme. This enables using three-phase voltages as real-valued control variables. Additionally, with modulation schemes a fixed switching frequency can be produced. If the switches are controlled directly, the modulation stage is omitted. With no modulator, the control variable is not a real-valued variable but an integer variable, which complicates control. With direct control, there is no guarantee of a fixed switching frequency. [5]

This thesis proposes an MPC algorithm that controls the switches of the inverter directly while providing a fixed switching frequency. This is done by setting the inverter legs to switch independently and one per sapling interval. The optimization variable of the underlaying optimization problem is a vector of times to switch for the inverter legs. The stator current squared root mean square (rms) error is approximated and minimized under certain assumptions.

This thesis consists of five chapters. Chapter 2 presents the theory behind MPC for ac drives. Chapter 3 outlines the proposed control method as well as its precursor algorithm. Chapter 4 displays the simulations that compare the different control methods discussed in the thesis. The differences and similarities are highlighted and analyzed. The last chapter, Chapter 5, draws the conclusions and suggestions for future research.

2. THEORY

This chapter presents the theoretical background required for this thesis. Key concepts behind motor control and MPC for ac drives are briefly presented. Some of the conventional control strategies are also covered.

Section 2.1 introduces the concept of stationary and rotating orthogonal reference frames and the transformation matrices associated with them. The state-space model of an induction machine (IM) is presented in Section 2.2. Section 2.3 presents the modeling of a two-level inverter. The concept of CB-PWM and the harmonics associated with it are in Section 2.4. Field oriented control (FOC) is briefly introduced in Section 2.5. Section 2.6 contains definitions and concepts related to mathematical optimization. The notion of MPC is introduced in Section 2.7. One MPC algorithm for electric drives, i.e., finite control set MPC (FCS-MPC), is presented in Section 2.8.

2.1 Orthogonal Reference Frames

In this thesis, three-phase quantities are transformed into independent quantities in orthogonal reference frames that are either stationary or rotating. This is done by employing Clarke and Park transformations. The system modeling and the controller design is simplified by employing the aforementioned transformations.

2.1.1 Stationary Orthogonal Reference Frame

The orthogonal stationary reference frame is also known as the $\alpha\beta 0$-coordinate system. Its axes are perpendicular to each other and do not rotate. The *abc* and $\alpha\beta$ coordinates are visualized in Fig. 2.1. Three-phase quantities can be expressed in $\alpha\beta 0$ coordinates, or vice versa, without any loss of information. To transform a three-phase quantity to $\alpha\beta 0$ coordinates, the Clarke transformation [6] is used.

The α and β components are differential-mode components, while the zero component is the common-mode component. By definition, a common-mode signal is the

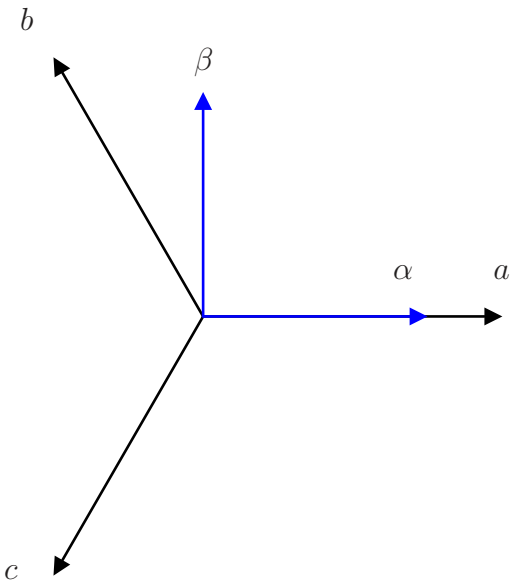

Figure 2.1 *abc and αβ coordinates*

same for all three phases. In case of a balanced, symmetrical three-phase system there is no common-mode component, thus the reduced Clarke transformation is sufficient:

$$\boldsymbol{K} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}. \tag{2.1}$$

The term $\frac{2}{3}$ in (2.1) is used to make the transformation amplitude invariant. This means that a three-phase signal with a certain amplitude has the same amplitude in the $\alpha\beta$ plane. The transformation matrix $\boldsymbol{K}$ is used to map the three-phase signal into the two-dimensional stationary plane.

To transform a quantity from $\alpha\beta$ to a three-phase quantity, the reduced inverse Clarke transformation is used. The inverse Clarke transformation matrix with no zero component is given by

$$\boldsymbol{K}^{-1} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}. \tag{2.2}$$

2.1.2 Rotating Orthogonal Reference Frame

The rotating orthogonal reference frame is also known as the $dq0$ coordinate system. Compared with the $\alpha\beta0$ plane, it rotates counterclockwise with an angular speed ω_{fr}. When transforming quantities from the $\alpha\beta$ to the dq plane, they are rotated

clockwise. Consequently, signals in the dq frame are dc quantities.

To rotate instantaneous values at an angular speed, it is necessary to use the angle φ, which is the angle between the d-axis and a-axis of the three-phase system. To transform $\alpha\beta$ quantities to the dq domain the rotational matrix

$$\boldsymbol{R}(\varphi) = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{bmatrix}, \tag{2.3}$$

is used. To transform dq quantities to $\alpha\beta$ quantities, the rotation is done in the opposite direction. Rotating dq quantities counterclockwise can be done with the rotation matrix

$$\boldsymbol{R}^{-1}(\varphi) = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix}. \tag{2.4}$$

Transforming a three-phase signal from abc to dq can be done by using the Park transformation [21]. The same results will be achieved by first using the Clarke transformation and then rotating the result by employing matrix $\boldsymbol{R}(\varphi)$, as explained above. The Park transformation and the reverse Park transformation are given in (2.5) and in (2.6), respectively.

$$\boldsymbol{K}(\varphi) = \frac{2}{3} \begin{bmatrix} \cos(\varphi) & \cos(\varphi - \frac{2\pi}{3}) & \cos(\varphi + \frac{2\pi}{3}) \\ -\sin(\varphi) & -\sin(\varphi - \frac{2\pi}{3}) & -\sin(\varphi + \frac{2\pi}{3}) \end{bmatrix} \tag{2.5}$$

$$\boldsymbol{K}^{-1}(\varphi) = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \cos(\varphi - \frac{2\pi}{3}) & -\sin(\varphi - \frac{2\pi}{3}) \\ \cos(\varphi + \frac{2\pi}{3}) & -\sin(\varphi + \frac{2\pi}{3}) \end{bmatrix}. \tag{2.6}$$

2.2 Induction Machine Model

This section presents the state-space model of an IM. Note that the derivation of the model is not presented since it is out of the scope of this thesis. The interested reader is referred to [18] for more details.

When deriving the state-space model of the IM, some simplifications are made. Mainly, machine parameters do not change based on heat, the inductor does not saturate and the machine is symmetrical.

The IM is modeled using the state vector $\boldsymbol{x} = [i_{s,\alpha}\ i_{s,\beta}\ \psi_{r,\alpha}\ \psi_{r,\beta}]^T$, the input vector $\boldsymbol{u} = [v_\alpha\ v_\beta]^T$ and the output vector $\boldsymbol{y} = [i_{s\alpha}\ i_{s\beta}]^T$, where $\boldsymbol{i}_s$ is the stator current, $\boldsymbol{\psi}_r$ is the rotor flux and $\boldsymbol{v}$ is the stator voltage. The continuous-time state-space model

is

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} &= \boldsymbol{F}\boldsymbol{x}(t) + \boldsymbol{G}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t), \end{aligned} \tag{2.7}$$

where $\boldsymbol{F}$ is the system matrix, $\boldsymbol{G}$ is the input matrix and $\boldsymbol{C}$ is the output matrix, which are as follows:

$$\boldsymbol{F} = \begin{bmatrix} \frac{-1}{\tau_s} & 0 & \frac{x_m}{\tau_r D} & \frac{\omega_r x_m}{D} \\ 0 & \frac{-1}{\tau_s} & \frac{-\omega_r x_m}{D} & \frac{x_m}{\tau_r D} \\ \frac{x_m}{\tau_r} & 0 & \frac{-1}{\tau_r} & -\omega_r \\ 0 & \frac{x_m}{\tau_r} & \omega_r & \frac{-1}{\tau_r} \end{bmatrix} \tag{2.8}$$

$$\boldsymbol{G} = \frac{x_r}{D} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{2.9}$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \tag{2.10}$$

where r_s, r_r, x_{ls}, x_{rs} and x_m are parameters of the IM in per unit; r_s (r_r) is the stator (rotor) resistance, and x_{ls} (x_{rs}) and x_m are the stator (rotor) and mutual leakage reactances. ω_r is the rotational speed of the rotor shaft.

The determinant D depends on the reactances (see (2.11)). The stator x_s and rotor x_r reactances are deduced from the leakage and mutual reactances, according to (2.12) and (2.13), respectively. Finally, the stator τ_s and rotor τ_r time constants are given by (2.14) and (2.15), respectively.

$$D = x_s x_r - x_m^2 \tag{2.11}$$

$$x_s = x_{ls} + x_m \tag{2.12}$$

$$x_r = x_{lr} + x_m \tag{2.13}$$

$$\tau_s = \frac{x_r D}{r_s x_r^2 + r_r x_m^2} \tag{2.14}$$

$$\tau_r = \frac{x_r}{r_r}. \tag{2.15}$$

For the MPC schemes discussed in this thesis, the continuous-time dynamic model of the IM has to be discretized based on the sampling interval T_s. The discrete-time state-space model is given in (2.16).

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ \boldsymbol{y}(k) &= \boldsymbol{C}\boldsymbol{x}(k), \end{aligned} \tag{2.16}$$

where $k \in \mathbb{N}$ and $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are the discrete-time state-space matrices. The output matrix $\boldsymbol{C}$ is the same as for the continuous-time model. The state and input matrices result from discretizing the model so their expression depends on the discretization method used. If the continuous-time model is discretized using exact Euler discretization, the resulting discrete-time matrices are

$$\boldsymbol{A} = \boldsymbol{e}^{\boldsymbol{F}T_s} \tag{2.17}$$

$$\boldsymbol{B} = -\boldsymbol{F}^{-1}(\boldsymbol{I} - \boldsymbol{A})\boldsymbol{G}, \tag{2.18}$$

where $\boldsymbol{e}$ denotes the matrix exponential and $\boldsymbol{I}$ is the identity matrix that has the same dimensions as $\boldsymbol{A}$.

Exact Euler method is more accurate, but can be computationally intensive. Forward Euler method can be used to reduce the computations required. This discretization corresponds to the linear terms of the exact discretization. Therefore, if the sampling time is small enough with respect to the time constants of the system, the linearization is sufficiently accurate. The resulting matrices are

$$\boldsymbol{A} = \boldsymbol{I} + \boldsymbol{F}T_s \tag{2.19}$$

$$\boldsymbol{B} = \boldsymbol{G}T_s. \tag{2.20}$$

The rated values of the IM are given in Table 2.1. The values of the parameters of the machine are given in Table 2.2. For the sake of completeness, the dc-link voltage is also in the table. The values are given in SI units and p.u. values. The modeling and simulation are done in the p.u. system. The p.u. system and the selected base values is introduced in Appendix A.

Table 2.1 *Rated values of the induction machine*

Key	**Value**
Line to line voltage	3300 V
Stator current	356 A
Stator frequency	50 Hz
Rotational speed	596 rpm
Power factor	0.809
Slip	6.67 %

2.3 Inverter

An illustration of an ac drive is given in Fig. 2.2. It consists of five main parts: the grid, the rectifier, the dc-link, the inverter and the electric machine. There can

Table 2.2 *Induction machine parameters*

Parameter	Symbol	Value	p.u. symbol	p.u. value
Stator resistance	R_s	57.61 mΩ	r_s	0.0108
Rotor resistance	R_r	48.89 mΩ	r_r	0.0091
Stator leakage reactance	X_{ls}	2.544 mH	x_{ls}	0.1493
Rotor leakage reactance	X_{lr}	1.881 mH	x_{lr}	0.1104
Mutual reactance	X_m	40.014 mH	x_m	2.3489
Number of pole pairs	p	5		
Dc-link voltage	V_{dc}	5200 V	V_{dc}	1.9299

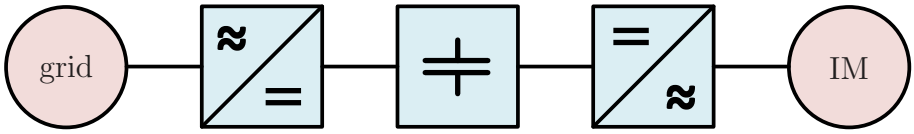

Figure 2.2 *Variable-speed drive system*

also be a transformer between the grid and the grid-side converter. The grid-side converter is used as a rectifier. It is used to keep the dc voltage at a set value even if the amount of current drawn from the capacitor changes. The dc link is a component with a large capacitance used to store energy and to decouple the two converters. The inverter converts dc to ac with variable frequency and amplitude.

Only the control of the electric machine is included in this thesis. To simplify the control problem in question the dc-link voltage is assumed to be constant.

As mentioned above, the conversion from dc to ac is done with an inverter. It consists of a set of switches that are switched on and off at specific time instants. The topology of the two-level inverter, which will be used in the thesis, is shown in Fig. 2.3.

The inverter has three legs, each controlling one of the phase voltages. Each leg has an upper and a lower switch. The upper and lower switches of a single leg cannot be on at the same time as this would effectively short-circuit the dc link. The switches are assumed ideal: they switch on or off instantly. Delays and dead time are neglected. This enables modeling the inverter using only the states of the switches of the three legs.

Depending on the state of an inverter leg, i.e., the position of its switches, the output phase voltage can be other V_{dc} or 0. The state-space model of the IM takes the phase voltages in $\alpha\beta$ as an input. Therefore, the input vector of the state-space model is

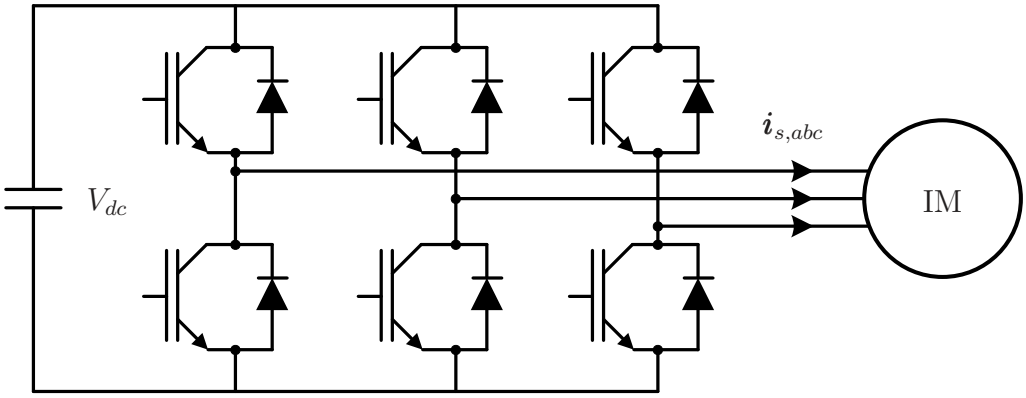

Figure 2.3 *Two-level inverter topology*

given by (2.21)

$$\boldsymbol{v} = V_{dc}\boldsymbol{K}\boldsymbol{u} \tag{2.21}$$

$$\boldsymbol{u} = \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix}, \tag{2.22}$$

where $u_x \in \{0, 1\}$ with $x \in \{a, b, c\}$. u_x is 1 if that leg provides V_{dc} as the phase voltage and 0 if the output is zero.

The inverter can produce a total of $2^3 = 8$ different switching combinations. When transformed to the $\alpha\beta$ plane, these combinations correspond to eight voltage vectors, six active and two zero voltage vectors. The possible voltage vectors in $\alpha\beta$-plane are illustrated in Fig. 2.4. [18]

2.4 Carrier Based Pulse Width Modulation

The control of a VSD is done by turning the switches of the inverter on and off. To employ linear control techniques the switching nature of the power converter has to be concealed. This is achieved by averaging the dynamics of the converters over the fundamental period. The voltage command that is generated by the adopted linear controllers is translated into switching commands with the help of a modulation stage, such as CB-PWM.

CB-PWM compares a reference signal, i.e., the desired voltage, with a carrier signal to determine the width of the generated pulses that are sent to the switches. The carrier signal is a triangular waveform of amplitude 1. The voltage reference signal

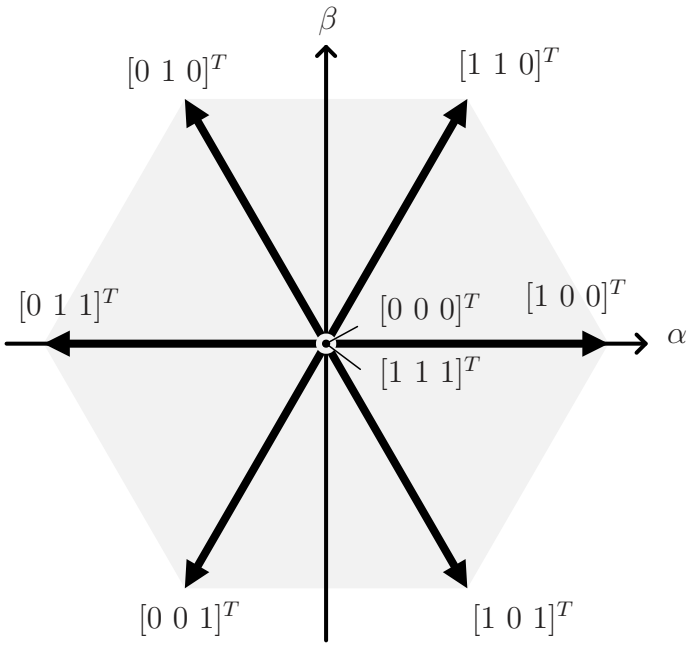

Figure 2.4 *Two-level inverter voltage vectors in the $\alpha\beta$ plane*

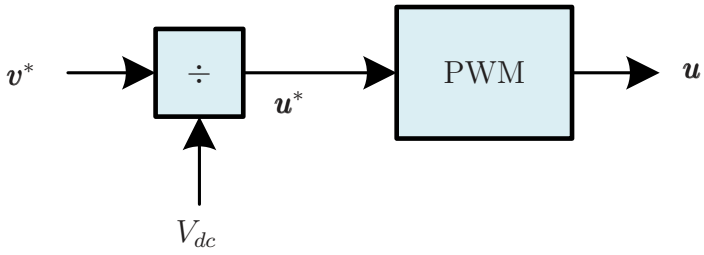

Figure 2.5 *Block diagram of PWM*

$\boldsymbol{v}^*$ has the form

$$\boldsymbol{v}^* = \hat{v}^* \begin{bmatrix} \sin(\omega t) \\ \sin(\omega t - 2\pi/3) \\ \sin(\omega t + 2\pi/3) \end{bmatrix}, \tag{2.23}$$

where ω is the angular speed and $\hat{v}^*$ is the amplitude of the voltage reference. $\boldsymbol{v}^*$ is produced by the current controllers of the control scheme that is utilizing CB-PWM. $\hat{v}^*$ is scaled with the dc-link voltage V_{dc}, which produces the CB-PWM reference signal $\boldsymbol{u}^*$ with amplitude $\hat{u}^* = \frac{\hat{u}^*}{V_{dc}}$. This process is illustrated in the block diagram seen in Fig. 2.5.

By comparing the carrier signal and $\boldsymbol{u}^*$, the phase voltage produced by the inverter will, on average, match the reference voltage. If $\boldsymbol{u}^*$ is greater than the carrier signal, the associated inverter leg is on and vice versa.

In steady-state operation, $\boldsymbol{v}^*$ is a symmetrical three-phase voltage where each phase

has a frequency of f_f and an amplitude of $\hat{v}^*$. f_f is the fundamental frequency. The ratio $m = \frac{\hat{v}^*}{V_{dc}}$ is the modulation index. With $m = 1$, the fundamental component of $\boldsymbol{u}$ is 1 p.u. If $m \leq 1$, one inverter leg switches twice per carrier period $T_c = \frac{1}{f_c}$ resulting in a predictable and constant switching frequency. If m is greater than one, CB-PWM enters overmodulation: the nonlinear regime where the linear relationship between the modulating signal $\boldsymbol{u}^*$ and the fundamental component of the output signal $\boldsymbol{u}$ ceases to exist.

The frequency of the carrier signal f_c can be selected arbitrarily, as long as $f_c \gg f_f$. In synchronous CB-PWM f_c is chosen as $f_c = (2(n+1) - 1)f_f$, where $n \in \mathbb{N}$. Synchronous CB-PWM causes the pulse pattern $\boldsymbol{u}(t)$ to repeat itself periodically for fundamental periods $T_f = \frac{1}{f_f}$. It also causes half-wave symmetry: $u_a(t - \frac{1}{2}T_f) = -u_a(t)$. The pulse pattern of the first half of the fundamental period repeats at the second half but mirrored.

When a pulse pattern produced in a quarter of the fundamental period is the same as the quarter next to it but mirrored, the pulses are quarter-wave symmetric, provided that there exists half-wave symmetry. This additional symmetry can be achieved by aligning the phase of the carrier signal and the reference signal in a specific way. This adds an additional restriction to the carrier frequency: f_c has to be triplen, odd integer multiple of f_f. In other words, $f_c = (3 + 6n)f_f$, where $n \in \mathbb{N}$. This enables aligning the phase difference between the carrier and all three reference signals.

CB-PWM can be implemented in analog or discrete manner. The discrete CB-PWM introduces a small delay. The delay can be compensated for by a phase offset in the carrier signal. Moreover, in discrete CB-PWM, the reference signal is sampled at regular intervals. The sampling can be done either at only the upper peaks of the carrier signal, or at both the upper and lower peaks. Sampling at both peaks, i.e., two times in a carrier period, is called asymmetric sampling. The delay caused by using asymmetric sampling is a quarter of one carrier period, i.e. $\frac{T_c}{4}$.

Asymmetrically sampled synchronous CB-PWM is illustrated in Fig. 2.6. The fundamental frequency f_f is 50 Hz and the carrier frequency f_c is 450 Hz. The carrier signal has an offset of $1.5\pi\frac{f_f}{f_c}$, so that the produced pulse pattern, seen in Fig. 2.7, is quarter-wave symmetric.

The harmonic spectrum of the pulse pattern seen in Fig. 2.7 is shown in Fig. 2.8. When using synchronous CB-PWM, the harmonics can be found according to

$$f_{\mu\nu} = \mu f_c + \nu f_f, \tag{2.24}$$

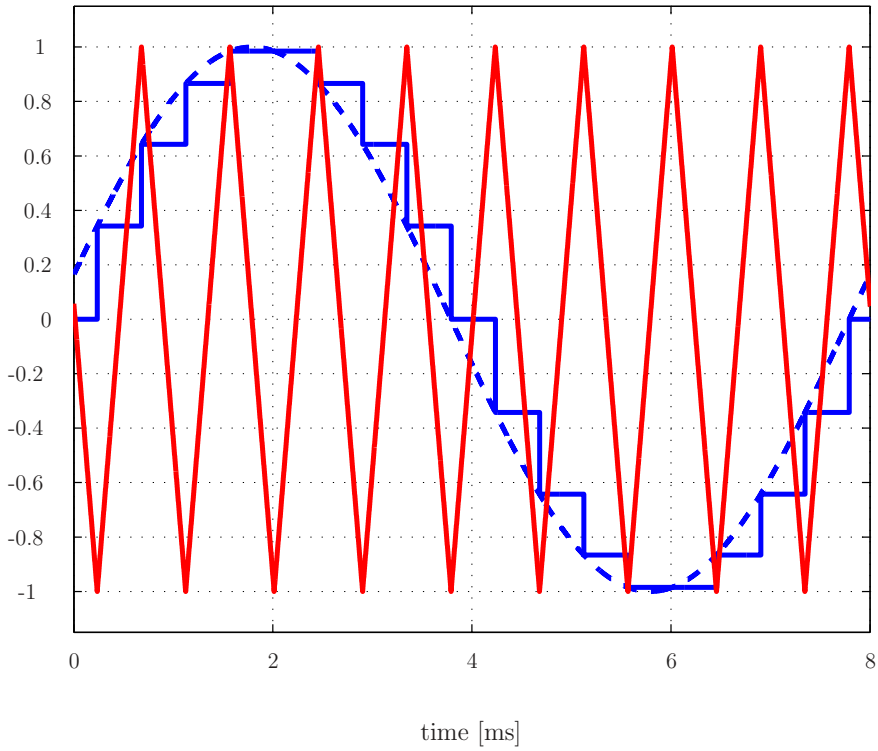

Figure 2.6 *Phase a of the reference signal (blue) compared to the carrier signal (red) with* $m = 1$

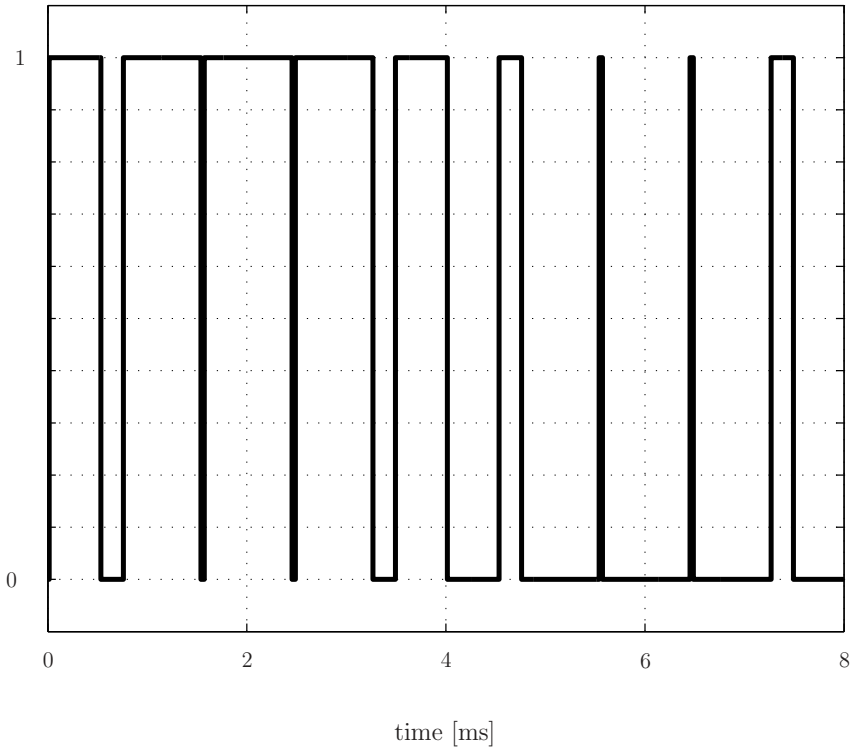

Figure 2.7 *The quarter-wave symmetric pulse pattern*

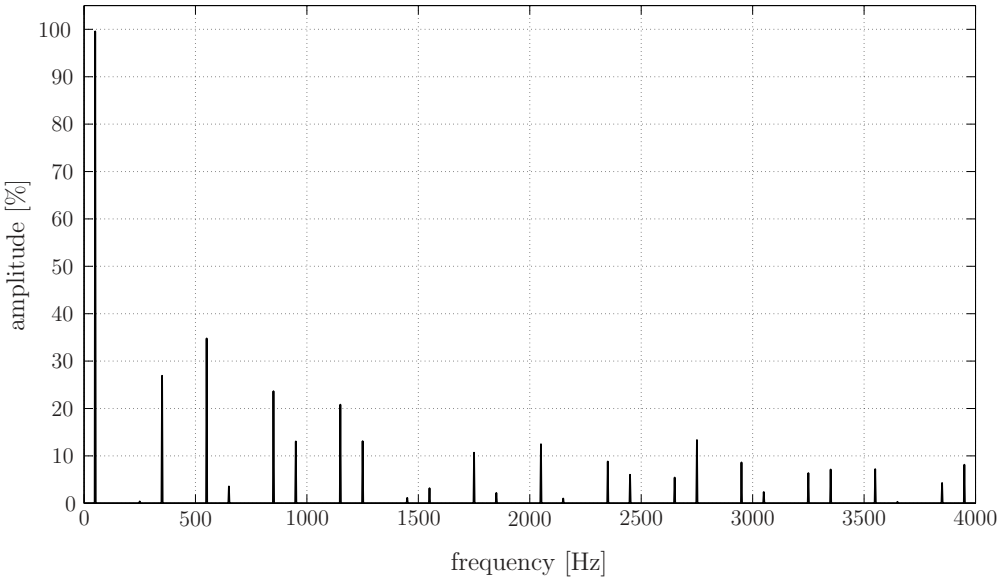

Figure 2.8 *Frequency components of* u_α

where $\mu \in \mathbb{N}$, $\nu \in \mathbb{Z}$ and for regularly sampled CB-PWM, $\mu + \nu = 2z + 1$, where $z \in \mathbb{Z}$. As can be understood from (2.24), harmonics at even multiples of the fundamental do not exist, owing to the quarter-wave symmetry of the pulse pattern. Additionally, as will be explained in the next section, a balanced three-phase system will not have triplen harmonics. [13]

2.4.1 Common-Mode Injection

An inverter produces three-phase ac quantities. If the three phases are symmetrical, there is no common-mode component on the machine side of the ac drive. This means that the harmonics that are triplen odd multiples of the fundamental do not exist. Additionally, this phenomenon can be utilized by the inverter to produce a fundamental component with amplitude greater than one. With common-mode injection, the reference signals amplitude can be increased up to $\frac{2}{\sqrt{3}}$ without going into overmodulation. Two common-mode voltage injection methods are presented: injecting a third harmonic or a min/max based signal.

In third harmonic injection, the reference signal is injected with a common-mode signal that has a frequency of $3f_f$ and an amplitude of $\frac{m}{6}$. As mentioned, the triplen harmonics do not exist. The peaks of the final reference signal get flatter while preserving the fundamental component.

In min/max based injection, the following common-mode signal is added to the reference signals: $-\frac{1}{2}(\min(\boldsymbol{u}^*_{abc}) + \max(\boldsymbol{u}^*_{abc}))$. With this method, the highest and

the lowest instantaneous values of the phases always have an equal distance from zero. This also flattens the produced reference signals. [13]

2.5 Field Oriented Control

FOC is a popular control scheme for ac drives. It achieves very good performance at steady-state operating conditions owing to the independent control of torque and flux.

The main idea in rotor flux oriented FOC is to perform the control in dq reference plane and to align the d-axis with the rotor flux vector $\boldsymbol{\psi}_r$. The use of orthogonal reference frames separates the stator current $\boldsymbol{i}_{s,abc}$ into d and q components. The d component controls the flux magnitude and the q component controls the torque.

A block diagram of rotor flux oriented FOC is given in Fig. 2.9 [8]. It contains two cascaded control loops with PI controllers. The inner loop controls the d and q stator currents while the outer loop controls the rotor flux magnitude and the torque. The inner loop has to be tuned to be sufficiently faster than the outer loop.

The current control block seen in Fig. 2.9 contains cross-coupling terms. The d and q components of the stator current are not perfectly decoupled during transients; the feed-forward terms given in (2.25) and (2.26) are added to the output of the controllers to decouple them.

$$v_{sd,ff} = -\omega_s X_\sigma i_{s,q} \tag{2.25}$$

$$v_{sq,ff} = \omega_s X_\sigma i_{s,d}, \tag{2.26}$$

where $\boldsymbol{v}_{s,ff}$ is the stator voltage feed-forward term, ω_s is the stator side rotational speed and $X_\sigma = \frac{D}{X_r}$ is the total leakage reactance.

The current controller produces the voltage reference. To translate it into switching signals a modulator is required. Here, synchronous asymmetric regularly sampled CB-PWM is used. [2][12]

2.6 Optimization

To properly introduce the basic concepts of mathematical optimization, three definitions are stated. A convex set, a convex function and an affine function are defined in (2.27), (2.28) and (2.29) respectively. These are directly from [3].

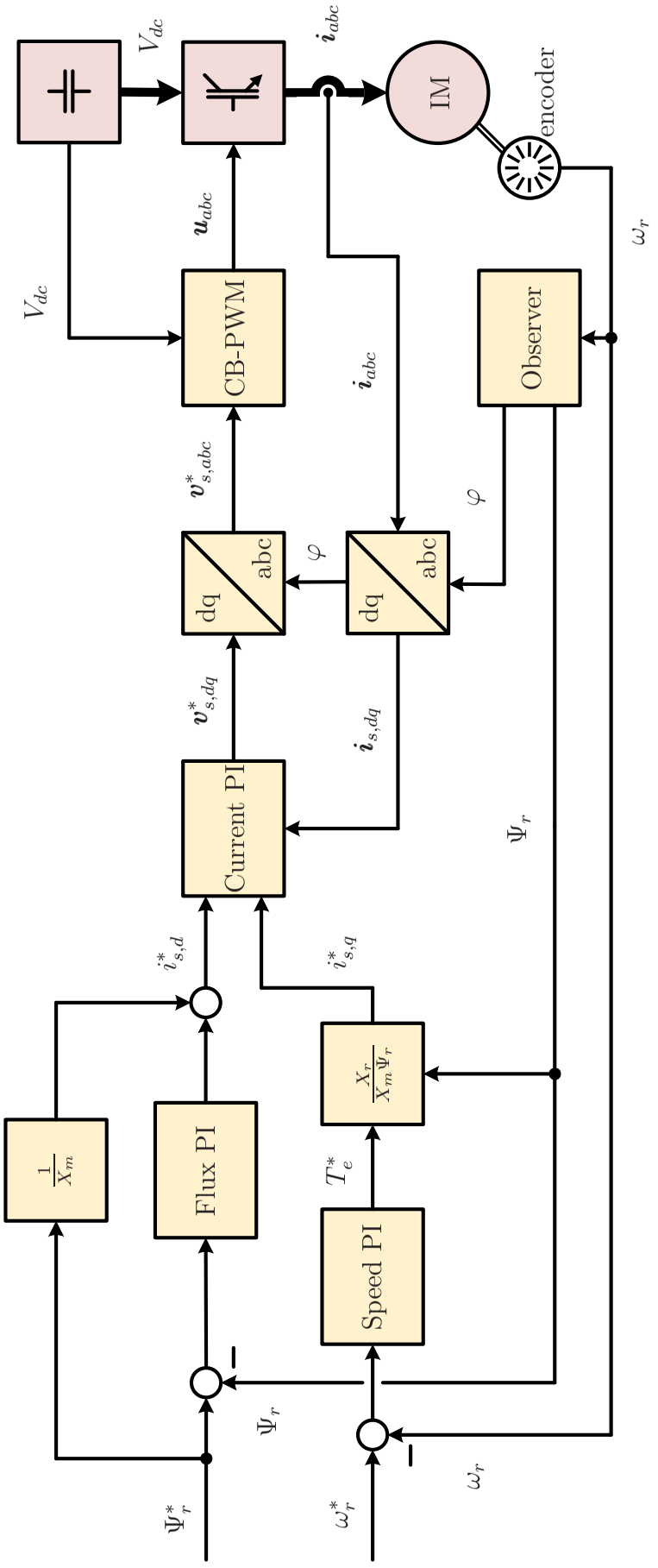

Figure 2.9 *Block diagram of FOC*

A set C is convex if the line segment between any two point in C lies in C, i.e., if for any $\boldsymbol{x}_1$, $\boldsymbol{x}_2 \in C$ and any θ with $0 \leq \theta \leq 1$ we have

$$\theta \boldsymbol{x}_1 + (1-\theta)\boldsymbol{x}_2, 0 \leq \theta \leq 1. \tag{2.27}$$

A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if $\mathbf{dom}\ f$ is a convex set and if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbf{dom} f$ and θ with $0 \leq \theta \leq 1$ we have

$$f(\theta \boldsymbol{x} + (1-\theta)\boldsymbol{y}) \leq \theta f(\boldsymbol{x}) + (1-\theta) f(\boldsymbol{y}). \tag{2.28}$$

A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is affine if it is a linear function plus a constant, i.e., if it has the form

$$f(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \tag{2.29}$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$.

The general form of an optimization problem is

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leq 0, \quad i = 1, ..., m \\ & h_j(\boldsymbol{x}) = 0, \quad j = 1, ..., p. \end{aligned} \tag{2.30}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the optimization variable, $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function, $g_i : \mathbb{R}^n \to \mathbb{R}$ are the inequality constraint functions and $h_j : \mathbb{R}^n \to \mathbb{R}$ are the equality constraint functions. If the objective function is being minimized, the term cost function is often used in its place.

The domain of the optimization problem $\mathcal{O}$ is defined as the intersection of domains of f, g and h, i.e.,

$$\mathcal{O} = \mathbf{dom} f \cap \bigcap_{i=1}^{m} \mathbf{dom} g_i \cap \bigcap_{j=1}^{p} \mathbf{dom} h_j. \tag{2.31}$$

A point $\boldsymbol{x} \in \mathcal{O}$ is feasible if it satisfies all the constraints $g_i(\boldsymbol{x}) \leq 0, i = 1, ..., m$ and $h_j(\boldsymbol{x}) = 0, j = 1, ..., p$. All feasible points form the feasible set $\mathbb{F}$. The optimization problem is feasible if $\mathbb{F} \neq \emptyset$ and infeasible if $\mathbb{F} = \emptyset$, where $\emptyset$ denotes the empty set.

The optimal value q^* is defined as the infimum of a set formed by the codomain of f where $\boldsymbol{x} \in \mathbb{F}$, i.e.,

$$q^* = \inf\{f(\boldsymbol{x}) \mid g_i(\boldsymbol{x}) \leq 0, i = 1, ..., m, h_j(\boldsymbol{x}) = 0, j = 1, ..., p\}. \tag{2.32}$$

The optimal point $\boldsymbol{x}^*$ is defined as a feasible point that results in the optimal value, i.e., $f(\boldsymbol{x}^*) = q^*$ and $\boldsymbol{x}^* \in \mathbb{F}$. All optimal points form the optimal set $\mathcal{X}_{\text{opt}}$, which is defined as

$$\mathcal{X}_{\text{opt}} = \inf\{\boldsymbol{x} \mid f(\boldsymbol{x}) = q^*, g_i(\boldsymbol{x}) \leq 0, i = 1, ..., m, h_j(\boldsymbol{x}) = 0, j = 1, ..., p\}. \tag{2.33}$$

Depending on the problem, $\mathcal{X}_{\text{opt}}$ can have zero, one or many elements. The problem is solvable if $\mathcal{X}_{\text{opt}} \neq \emptyset$.

2.6.1 Convex Optimization

Convex optimization is a subset of mathematical optimization where the problems are convex. There are many algorithms for solving convex problems efficiently. They can be solved in $O(n^2)$ time if certain conditions are met [20].

A convex problem is generally formulated as follows:

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leq 0, \quad i = 1, ..., m \\ & \boldsymbol{a}_j^T \boldsymbol{x} = b_j, \quad j = 1, ..., p, \end{aligned} \tag{2.34}$$

where $f(\boldsymbol{x})$ and $g_i(\boldsymbol{x}) \leq 0$ are convex and the equality constraint functions $\boldsymbol{a}_j^T \boldsymbol{x} = b_j$ are affine. This guarantees that $\mathcal{O}$ is a convex set. Additionally, $\mathcal{X}_{\text{opt}}$ is either empty or singleton.

2.6.2 Quadratic Optimization

As shown in Section 3.2., the optimization problem encountered in this thesis can be formulated as a convex quadratic program (QP). Such a problem can be written as

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{p}^T \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{G}\boldsymbol{x} \preceq \boldsymbol{h} \\ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \end{aligned} \tag{2.35}$$

where $\boldsymbol{Q} \in \mathbb{S}_+^n$, $\boldsymbol{p} \in \mathbb{R}^n$, $\boldsymbol{G} \in \mathbb{R}^{m \times n}$, $\boldsymbol{h} \in \mathbb{R}^m$, $\boldsymbol{A} \in \mathbb{R}^{p \times n}$ and $\boldsymbol{b} \in \mathbb{R}^p$. $\mathbb{S}_+$ is defined as symmetric and positive semidefinite. A positive (semi)definite $\boldsymbol{Q}$ implies that the problem is solvable in polynomial time [17].

2.6.3 Integer Problems

An (mixed) integer problem is an optimization problem where (some) elements of the optimization variable $\boldsymbol{x}$ are integers. The (mixed) integer optimization variable makes the problem non-convex. This, in turn, means that the problem is NP-hard and not solvable in polynomial time [26].

A linear mixed integer problem is defined as

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{G}\boldsymbol{x} \preceq \boldsymbol{h} \\ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x}_z \in \mathbb{Z}^{n_z}, \end{aligned} \tag{2.36}$$

where $\boldsymbol{x} = [\boldsymbol{x}_r^T \ \boldsymbol{x}_z^T]^T$, with $\boldsymbol{x}_r \in \mathbb{R}^{n_r}$, $\boldsymbol{x}_z \in \mathbb{Z}^{n_z}$ and $n = n_r + n_z$. In addition, $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{G} \in \mathbb{R}^{m \times n}$, $\boldsymbol{h} \in \mathbb{R}^m$, $\boldsymbol{A} \in \mathbb{R}^{p \times n}$ and $\boldsymbol{b} \in \mathbb{R}^p$.

If $n_r = 0$, i.e., the optimization variable consists only of integers, the problem is an integer problem.

2.7 Model Predictive Control

In a control problem, a plant is desired to behave in a certain way. To achieve this, the inputs of said plant are chosen or generated in a way that makes the plant to behave as desired. There are many ways this can be done, one of which is MPC.

In MPC, the plant needs to be modeled using a discrete-time state-space model. The model produces the states and outputs for the next discrete-time step as a function of the previous state variables and the input variables that are applied. The linear discrete-time state-space model has the form

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ \boldsymbol{y}(k+1) &= \boldsymbol{C}\boldsymbol{x}(k+1), \end{aligned} \tag{2.37}$$

where $\boldsymbol{x}$ is the state vector, $\boldsymbol{u}$ is the input vector and $\boldsymbol{y}$ is the output vector. This model is used to predict the evolution of the state and output variables over a number of steps. The number of discrete-time steps included in the prediction is called prediction horizon length and it is denoted by $N_P \in \mathbb{N}$. Additionally, the time window within which the system evolution is computed is called the prediction horizon.

The task of having the plant behave a certain way is addressed in the form of an optimization problem. To achieve the desired plant behavior an optimization problem is formulated. The abstract control objectives are expressed using the cost:

$$\boldsymbol{J}(\boldsymbol{x}(k), \boldsymbol{U}(k)) = \sum_{l=k}^{k+N_P+1} \Lambda(\boldsymbol{x}(l), \boldsymbol{u}(l)), \tag{2.38}$$

where $\Lambda(\cdot, \cdot)$ gives the cost for each time step. The control objectives, e.g., output reference tracking, are mapped to the cost function in such a way that a higher cost corresponds to a less desirable output. In other words, the cost function is used to penalize unwanted behavior by assigning cost to it. The cost function maps the evolution of the states and inputs over the prediction horizon to a non-negative scalar.

As explained in Section 2.6, an optimization problem can have any number of constraints. In MPC, the problem is constrained by the state-space model equations. It should be noted that there can be any number of additional equality or inequality constraints restricting any of the concerned variables [8].

To express the control actions within the prediction horizon the sequence of input vectors $\boldsymbol{U}(k)$ is defined:

$$\boldsymbol{U}(k) = [u^T(k)\ u^T(k+1)\ \ldots\ u^T(k+N_P-1)]^T. \tag{2.39}$$

Solving the optimization problem yields two things: the optimal sequence of control actions and the cost associated with that sequence. The optimal sequence of control actions $\boldsymbol{U}_{\text{opt}}$ is the sequence which produces the lowest cost while adhering to the limitations expressed by the constraints. The general form of the constrained optimization problem being solved is

$$\begin{aligned} \underset{\boldsymbol{U}(k)}{\text{minimize}} \quad & \boldsymbol{J}(\boldsymbol{x}(k), \boldsymbol{U}(k)) \\ \text{subject to} \quad & \boldsymbol{x}(l+1) = \boldsymbol{A}\boldsymbol{x}(l) + \boldsymbol{B}\boldsymbol{u}(l) \\ & \boldsymbol{y}(l+1) = \boldsymbol{C}\boldsymbol{x}(l+1) \\ & \forall l = k, ..., k+N_P-1. \end{aligned} \tag{2.40}$$

$\boldsymbol{U}_{\text{opt}}(k)$ gives an open loop solution to the control problem at time step k. To achieve closed loop control, only the first element of $\boldsymbol{U}_{\text{opt}}(k)$ is applied. At the next time step, the optimization problem is solved again. This is called the receding horizon policy. More specifically, at each time step, the controlled variables are measured

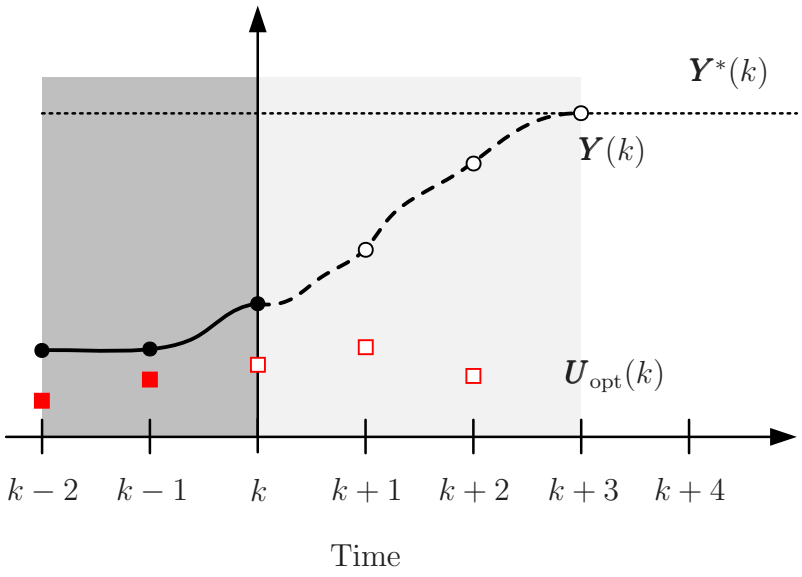

Figure 2.10 *Receding horizon visualization, time-step k*

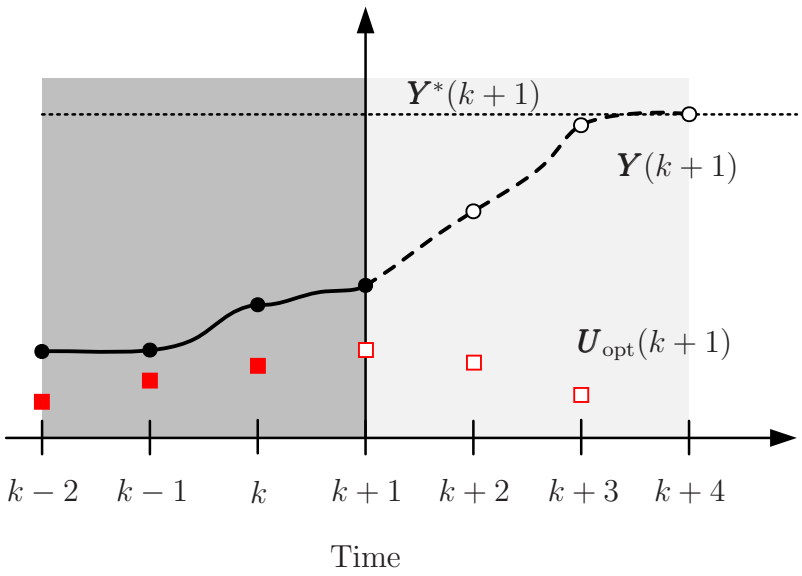

Figure 2.11 *Receding horizon visualization, time-step $k+1$*

and/or estimated. The control problem is solved using the measured/observed values, thus, even in the presence of model mismatches, unmodeled dynamics, etc., the performance of the controller is not significantly affected. At each time step, the horizon is shifted one step further, meaning that the horizon is always of length N_P time steps. Receding horizon with $N_P = 3$ is visualized in Fig. 2.10 and Fig. 2.11.

Even though only the inputs for the next step are applied, a longer prediction horizon is beneficial. When $\boldsymbol{U}_{\text{opt}}$ is solved, the cost function considers the behavior of the system along the whole prediction horizon. As a consequence, a longer prediction

horizon means that the behavior of the system is predicted further into the future. In the context of minimizing the cost function over time, when a longer time window is considered, better educational choices can be made. Thus, having a longer prediction horizon means that the algorithm is less aggressive.

As demonstrated in [9], MPC for ac drives benefits from a long prediction horizon, especially at low switching frequencies. However, if $\boldsymbol{U}$ has integer variables, the computational complexity required greatly (i.e., exponentially) increases as N_P gets longer. [8]

MPC can be summarized in the following steps:

1. Do the necessary measurements/observations
2. Predict behavior over the horizon
3. Solve the optimization problem
4. Get the sequence of inputs that produces the optimal result
5. Apply the inputs, but only for the first time step
6. Shift the prediction horizon by one time step
7. Repeat.

2.8 Finite Control Set Model Predictive Control

FCS-MPC is an MPC algorithm that can be used to control a signal when the input vector has integer variables. Although the algorithm is presented in a general form, the name FCS-MPC is given to it by the power electronic community. Thus, FCS-MPC refers to a specific MPC algorithm when used in power electronic applications.

In FCS-MPC, the signal(s) is (are) regularly sampled at intervals of T_s. Based on the discrete-time model of the plant, the states and outputs are predicted N_P time steps into the future. The optimization problem is solved, yielding the sequence of optimal inputs. The inputs are kept constant within one sampling interval. Finally, receding horizon policy is applied.

In FCS-MPC with $N_P = 1$, the cost function is

$$J(k) = \|\boldsymbol{s}^*(k+1) - \boldsymbol{s}(k+1)\|_2^2 + \lambda\|\Delta\boldsymbol{u}(k+1)\|_2^2, \tag{2.41}$$

where $\boldsymbol{s}$ is the controlled signal, $\boldsymbol{s}^*$ its reference value, $\|\cdot\|_2^2$ is the 2-norm, $\lambda \in \mathbb{R}^+$ is the tuning parameter and $\|\Delta\boldsymbol{u}(k+1)\|_2^2 = \|\boldsymbol{u}(k+1)-\boldsymbol{u}(k)\|_2^2$ is a term that penalizes changes in the control effort. In the context of power electronics this term can be referred as the switching effort.

The first part of the cost function, $\|\boldsymbol{s}^*(k+1) - \boldsymbol{s}(k+1)\|_2^2$, penalizes the error (as quantified by the Euclidean distance) between the controlled signals and their references. The 2-norm means that the error is squared, which results in a big cost for bigger errors. The reasons to use the 2-norm in the cost function are detailed in [14].

The second part of the cost function, $\|\Delta\boldsymbol{u}(k+1)\|_2^2$, consists of the switching effort. By penalizing changes in the control effort, one can minimize the switching transitions. Thus, owing to the second term one can in effect control the switching frequency.

The weighting factor λ gives the relative importance between the tracking error term and the switching effort. A higher λ corresponds to more emphasis on switching effort and vice versa. It is possible to set λ to be zero, which is equivalent to ignoring the switching effort completely.

The cost function in (2.41) is given for $N_P = 1$. In FCS-MPC, the input variables are integer variables, which makes the optimization problem an integer problem. As mentioned earlier, such problems are computationally demanding ($O(c^n)$) [23]. As a consequence, only FCS-MPC with N_P of 1 is considered in this thesis. For the interested reader, methods to reduce the computation required when using FCS-MPC with long prediction horizons are given in [10], [15] and [16].

The optimization problem for FCS-MPC is

$$\begin{aligned} \underset{\boldsymbol{u}}{\text{minimize}} \quad & J(k) \\ \text{subject to} \quad & \boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ & \boldsymbol{y}(k+1) = \boldsymbol{C}\boldsymbol{x}(k+1). \end{aligned} \tag{2.42}$$

One way to solve (2.42) is enumeration: all possible input combinations are evaluated before selecting the input that provides the most desirable result. There are a^b different input combinations, where a is the cardinality of the input set and b is the number of input variables. $\mathcal{U}$ is a set defined to contain all a^b possible input combinations. Enumeration for $N_P = 1$ and $\boldsymbol{s} = \boldsymbol{y}$ is given in Algorithm 1. It is executed at every time step.

Algorithm 1 FCS-MPC Enumeration, $N_P = 1$

$\text{cost}_{\min} \leftarrow \infty$
for $\boldsymbol{u} \in \mathcal{U}$ **do**
  $\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k)$
  $\boldsymbol{y}(k+1) = \boldsymbol{C}\boldsymbol{x}(k+1)$
  $\text{cost} \leftarrow J(k) = \|\boldsymbol{y}^*(k+1) - \boldsymbol{y}(k+1)\|_2^2 + \lambda\|\Delta\boldsymbol{u}(k+1)\|_2^2$
  if $\text{cost} < \text{cost}_{\min}$ **then**
    $\text{cost}_{\min} \leftarrow \text{cost}$
    $\boldsymbol{u}_{\text{opt}} \leftarrow \boldsymbol{u}$
  end if
end for

For current control of ac drives, the controlled signal $\boldsymbol{s}$ is replaced by $\boldsymbol{i}_s$. Then, λ can be used to scale the importance of a low current error versus a low switching effort. [23]

3. DIRECT MODEL PREDICTIVE CONTROL WITH FIXED SWITCHING FREQUENCY

3.1 Variable Switching Point Predictive Current Control

In FCS-MPC, switching can only happen at time-steps $k+n$, where $n \in \mathbb{N}$. The best of the eight possible switch positions is chosen and applied for a duration of T_s. VSP^{2}CC extends this principle by allowing the change of switch position to be delayed. Doing so enables both reference tracking and minimizing the ripple of the controlled signal.

VSP^{2}CC consists of three parts:

1. Calculating slopes
2. Finding the optimal time to switch
3. Solving the optimization problem

These are explained in the following sections.

3.1.1 Part 1: Calculating Slopes

In order to find the optimal time to switch, i.e., part 2, the slopes of the stator current need to be estimated. In VSP^{2}CC, a slope is estimated for each of the eight switch positions. The slopes are denoted with $\boldsymbol{m} = [m_\alpha \; m_\beta]^T$, where $m_{\alpha(\beta)} = \frac{\Delta i_{s,\alpha(\beta)}}{\Delta t}$.

The stator current slopes are estimated using the discrete-time state-space model of the IM given in (2.37). The continuous-time model is discretized using a sampling interval of T_s, which is the sampling interval for the controller. Then, based on the known state variable $\boldsymbol{x}(k)$, and by setting a particular switch position $\boldsymbol{u}(k)$ as an input, the future values of the stator currents are computed by $\boldsymbol{i_s}(k+1) = \boldsymbol{C}(\boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k))$. These predicted currents result if that particular switch position

$\boldsymbol{u}(k)$ were to be applied for the duration of one sampling interval. Now, the stator currents at time-step k and at $k+1$ are known along with the duration between these steps. These are used to calculate the slopes.

When the slopes are calculated, the stator currents are assumed to behave linearly, i.e., to have a constant rate of change during the sampling interval. This is not how the stator currents of an induction machine behave, but with a small sampling interval, the mismatch is relatively small.

The slopes for a particular switch position $\boldsymbol{u}(k)$ are found as follows:

$$\boldsymbol{m} = \frac{\boldsymbol{i}_s(k+1) - \boldsymbol{i}_s(k)}{T_s}, \tag{3.1}$$

where $\boldsymbol{i}_s(k+1)$ is found by applying $\boldsymbol{u}(k)$ for the duration of the whole sampling interval.

3.1.2 Part 2: Optimal Time to Switch

After part 1, each of the eight possible three-phase switch positions $\boldsymbol{u}$ is associated with a constant stator current slope. In part 2, the slopes are used to find the optimal time to switch for each of the switch positions.

As mentioned earlier, in VSP^{2}CC, it is possible to delay switching from a switch position to another. The amount of delay is called the optimal time to switch, $t_{\text{sw}} \in [0, T_s)$. In terms of time-steps, the switching instance is $n_{\text{sw}} \in [k, k+1)$. The slope associated with the switch position being applied at time 0 is denoted by $\boldsymbol{m}_0$. After the delay, a new switch position is applied. The slope associated with that switch position is denoted by $\boldsymbol{m}_1$.

For a particular switch position, t_{sw} is found by minimizing the root mean squared (rms) error of the stator current. The stator current squared rms error e_{rms} is by definition

$$e_{rms} = \frac{1}{T_s}\left(\int_0^{t_{\text{sw}}} (\boldsymbol{i}_s^* - \boldsymbol{i}_s(t))^2 dt + \int_{t_{\text{sw}}}^{T_s} (\boldsymbol{i}_s^* - \boldsymbol{i}_s(t))^2 dt\right). \tag{3.2}$$

To find the value for t_{sw} that minimizes the squared rms error, some assumptions need to be made. The stator current reference $\boldsymbol{i}_s^*$ is assumed to be constant. Moreover, as mentioned in Section 3.1.1., the stator current evolves linearly with constant slope $\boldsymbol{m}$, as given by (3.1) The progression of the stator currents with these assumptions in place is visualized in Fig. 3.1.

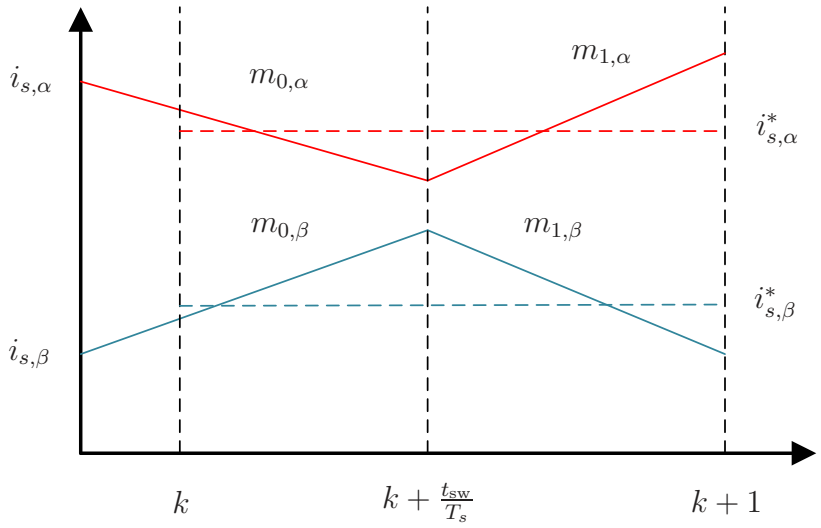

Figure 3.1 *Stator current slopes and the time to switch visualized*

With these assumptions, e_{rms} can be minimized by setting its derivative equal to zero. Then, after some calculus, minimizing e_{rms} takes the following form [24]:

$$t_{\text{sw}} = \frac{A + B}{C + D}, \tag{3.3}$$

where

$$\begin{aligned} A &= (m_{1,\alpha} - m_{0,\alpha})(2i_{s,\alpha}(0) - 2i^*_{s,\alpha} + T_s m_{1,\alpha}) \\ B &= (m_{1,\beta} - m_{0,\beta})(2i_{s,\beta}(0) - 2i^*_{s,\beta} + T_s m_{1,\beta}) \\ C &= (m_{0,\alpha} - m_{1,\alpha})(2m_{0,\alpha} - m_{1,\alpha}) \\ D &= (m_{0,\beta} - m_{1,\beta})(2m_{0,\beta} - m_{1,\beta}). \end{aligned} \tag{3.4}$$

3.1.3 Part 3: Solving the Optimization Problem

After part 2, each potential switch position is associated with an optimal time to switch. In other words, there are eight input pairs of the form $\{\boldsymbol{u}, t_{\text{sw}}\}$. In part 3, the pair that produces the most desirable results is chosen. This is formulated as an optimization problem.

The cost function for this optimization problem is chosen similarly to FCS-MPC, i.e., it consists of two terms: the tracking error term given by the squared 2-norm of the current error at switching and sampling instances and the switching effort term,

$$J(k) = \sum_{\xi \in S} (\|\boldsymbol{i}^*_s - \boldsymbol{i}_s(k + \xi|k)\|_2^2) + \lambda \|\Delta \boldsymbol{u}(k+1)\|_2^2, \tag{3.5}$$

where $S = \{\frac{t_{\text{sw}}}{T_s}, 1\}$. The term λ is a tuning parameter used to choose the relative importance of the switching effort compared to the current error.

The cost function requires finding $\boldsymbol{i}_s(t_{\text{sw}})$. This is done by discretizing the model for t_{sw} and applying the previous input $\boldsymbol{u}(k^-)$. Doing so yields $\boldsymbol{i}_s(t_{\text{sw}})$ and $\boldsymbol{x}(t_{\text{sw}})$. $\boldsymbol{i}_s(T_s)$ is found by using $\boldsymbol{x}(t_{\text{sw}})$ as the initial state, discretizing for $T_s - t_{\text{sw}}$ and applying the next switch position as an input.

The optimization problem for VSP^{2}CC is

$$\begin{aligned} \underset{\boldsymbol{u}}{\text{minimize}} \quad & J(k) \\ \text{subject to} \quad & \text{(3.1) and (3.3).} \end{aligned} \tag{3.6}$$

The optimization problem for VSP^{2}CC is a mixed integer problem. One way to solve it is to use exhaustive enumeration of all possible pairs $\{\boldsymbol{u}, t_{\text{sw}}\}$. Enumeration for VSP^{2}CC with $N_P = 1$ is given in Algorithm 2. [1][24]

Algorithm 2 VSP^{2}CC Enumeration, $N_P = 1$

```
cost_min ← ∞
get m_0
for u ∈ U do
    get m_1
    get t_sw
    discretize for t_sw
    get i_s(t_sw)
    discretize for T_s − t_sw
    get i_s(T_s)
    cost ← J(k)
    if cost < cost_min then
        cost_min ← cost
        {u_opt, t_sw,opt} ← {u, t_sw}
    end if
end for
apply u_opt at time t_sw,opt
repeat
```

3.2 Direct MPC With Fixed Switching Frequency

An MPC algorithm, direct MPC with fixed switching frequency, is proposed in this section. As the name suggests, the proposed method achieves a fixed switching frequency while controlling the switches directly, i.e. not using a modulator. This

is done by extending VSP^{2}CC to include more intermediate switching times, but simultaneously adding more constraints to the problem.

Similarly to VSP^{2}CC, direct MPC with fixed switching frequency has three parts:

1. Calculating slopes
2. Finding the optimal times to switch
3. Solving the optimization problem

These parts along with the algorithm specific constraints and definitions are presented in this section.

3.2.1 Constraints and Definitions

The average switching frequency of a two level inverter, f_{sw}, can be calculated by

$$f_{\text{sw}} = \lim_{N\to\infty} \frac{1}{6NT_s} \sum_{\ell=0}^{N-1} \|\boldsymbol{u}(\ell) - \boldsymbol{u}(\ell-1)\|_1, \tag{3.7}$$

where $N \in \mathbb{N}$ is the number of sampling intervals, 6 is the number of active switches and $\|\cdot\|_1$ denotes the 1-norm.

To achieve a constant switching frequency, each leg is forced to switch once within one T_s. This can be stated formally as

$$\|\boldsymbol{u}(k+1) - \boldsymbol{u}(k)\|_1^2 = 3. \tag{3.8}$$

With this restriction in place, the average switching frequency f_{sw} according to (3.7) is $\frac{1}{2T_s}$.

Another constraint is that each inverter leg is set to switch individually. In other words, two inverter legs cannot switch at the same time.

When these two constraints are combined, a total of four different switch positions are applied within one sampling interval. For example: let $\boldsymbol{u}(k) = [0\ 0\ 0]^T$. As each inverter leg has to switch once, it follows that $\boldsymbol{u}(k+1) = [1\ 1\ 1]^T$. Then, let leg a switch first, b second and c third. Because the legs are forced to switch individually, the following switch positions are applied: $[0\ 0\ 0]^T \to [1\ 0\ 0]^T \to [1\ 1\ 0]^T \to [1\ 1\ 1]^T$.

In the previous example, the order in which the inverter legs switched was assumed to be $a \to b \to c$. There are a total of $3! = 6$ different orders to switch in. The six switching orders and the resulting switch positions are given for $\boldsymbol{u}(k) = [0\ 0\ 0]^T$ in Table 3.1.

Table 3.1 *The six switching orders and the resulting switch positions for* $\boldsymbol{u}(k) = [0\ 0\ 0]^T$

Switching order	**Voltage vectors**
$a \to b \to c$	$\begin{bmatrix}0\\0\\0\end{bmatrix} \to \begin{bmatrix}1\\0\\0\end{bmatrix} \to \begin{bmatrix}1\\1\\0\end{bmatrix} \to \begin{bmatrix}1\\1\\1\end{bmatrix}$
$a \to c \to b$	$\begin{bmatrix}0\\0\\0\end{bmatrix} \to \begin{bmatrix}1\\0\\0\end{bmatrix} \to \begin{bmatrix}1\\0\\1\end{bmatrix} \to \begin{bmatrix}1\\1\\1\end{bmatrix}$
$b \to a \to c$	$\begin{bmatrix}0\\0\\0\end{bmatrix} \to \begin{bmatrix}0\\1\\0\end{bmatrix} \to \begin{bmatrix}1\\1\\0\end{bmatrix} \to \begin{bmatrix}1\\1\\1\end{bmatrix}$
$b \to c \to a$	$\begin{bmatrix}0\\0\\0\end{bmatrix} \to \begin{bmatrix}0\\1\\0\end{bmatrix} \to \begin{bmatrix}0\\1\\1\end{bmatrix} \to \begin{bmatrix}1\\1\\1\end{bmatrix}$
$c \to a \to b$	$\begin{bmatrix}0\\0\\0\end{bmatrix} \to \begin{bmatrix}0\\0\\1\end{bmatrix} \to \begin{bmatrix}1\\0\\1\end{bmatrix} \to \begin{bmatrix}1\\1\\1\end{bmatrix}$
$c \to b \to a$	$\begin{bmatrix}0\\0\\0\end{bmatrix} \to \begin{bmatrix}0\\0\\1\end{bmatrix} \to \begin{bmatrix}0\\1\\1\end{bmatrix} \to \begin{bmatrix}1\\1\\1\end{bmatrix}$

It should be noted that regardless of which switch position is being applied at k, the constraints cause six switching orders and four switch positions per switching order to appear.

For a particular switching order, switching times are defined as follows:

- $t_{\mathrm{sw},1} \in [0, T_s)$ is the time when the first inverter leg is switched
- $t_{\mathrm{sw},2} \in [0, T_s)$ is the time when the second inverter leg is switched
- $t_{\mathrm{sw},3} \in [0, T_s)$ is the time when the third inverter leg is switched

This enables stating the second constraint formally:

$$0 < t_{\mathrm{sw},1} < t_{\mathrm{sw},2} < t_{\mathrm{sw},3} < T_s. \tag{3.9}$$

A vector containing all three switching times is defined as

$$\boldsymbol{t}_{\text{sw}} = \begin{bmatrix} t_{\text{sw},1} \\ t_{\text{sw},2} \\ t_{\text{sw},3} \end{bmatrix}. \tag{3.10}$$

3.2.2 Part 1: Calculating Slopes

The first part in direct MPC with fixed switching frequency is to estimate stator current slopes. Each of the eight switch positions that the inverter can produce gets corresponding stator current slopes, exactly as in VSP^{2}CC. The method for estimating these slopes is also exactly as in VSP^{2}CC.

The stator current slope estimation method is repeated here for the sake of completeness:

$$\boldsymbol{m} = \frac{\boldsymbol{i}_s(k+1) - \boldsymbol{i}_s(k)}{T_s}, \tag{3.11}$$

where $\boldsymbol{i}_s(k+1)$ is found by applying the switch position in question for the duration of the whole sampling interval.

3.2.3 Part 2: Finding the Optimal Times to Switch

In VSP^{2}CC, each of the eight switch positions get associated with an optimal time to switch. The optimal switching time instant is found by finding the t_{sw} that minimizes the squared root mean square stator current error.

In direct MPC with fixed switching frequency, each of the six switching orders is associated with three switching times. By definition, the squared rms current error with three intermediate switching time instants is given by

$$\begin{aligned} e_{rms} = &\frac{1}{T_s}\int_0^{t_{\text{sw},1}} (\boldsymbol{i}_s^* - \boldsymbol{i}_s(t))^2 dt + \frac{1}{T_s}\int_{t_{\text{sw},1}}^{t_{\text{sw},2}} (\boldsymbol{i}_s^* - \boldsymbol{i}_s(t))^2 dt \\ &+\frac{1}{T_s}\int_{t_{\text{sw},2}}^{t_{\text{sw},3}} (\boldsymbol{i}_s^* - \boldsymbol{i}_s(t))^2 dt + \frac{1}{T_s}\int_{t_{\text{sw},3}}^{T_s} (\boldsymbol{i}_s^* - \boldsymbol{i}_s(t))^2 dt. \end{aligned} \tag{3.12}$$

In VSP^{2}CC, when assuming constant slopes and references, the problem of minimizing e_{rms} was solved by using calculus. The optimal switching time was expressed as a function of the initial current and the slopes. This is not possible for direct MPC with fixed switching frequency as the introduction of multiple intermediate

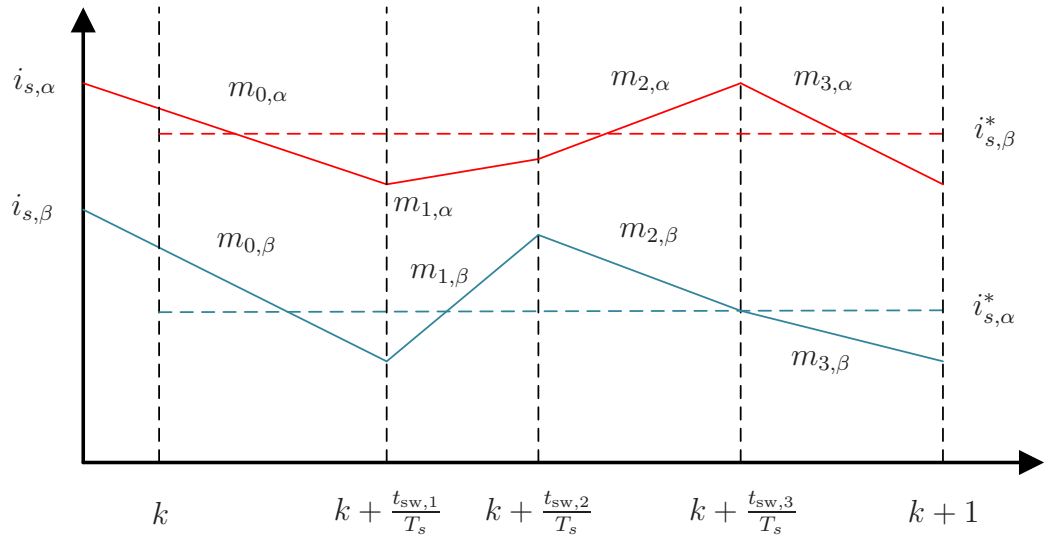

Figure 3.2 *Stator current slopes and the times to switch visualized*

switching times complicates the problem. Instead, the optimal switching times are calculated by minimizing the squared error at switching times. This simplification is a coarse approximation of the rms error. However, it enables formulating the problem of finding the three switching times as a convex optimization problem, provided that certain assumptions are made, as will be shown later.

When finding the optimal times to switch, $\boldsymbol{t}_{\mathrm{sw}}$, certain assumptions are made. The assumptions are the same as for VSP^{2}CC, i.e., constant stator current references and constant stator current slopes. As explained previously, four different switch positions get applied within one sampling interval. The constant slopes that correspond to the first switch position is defined as $\boldsymbol{m}_0$. For the second, third and fourth, the slopes are $\boldsymbol{m}_1$, $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$ respectively.

With the help of the previous definitions, the stator current is approximated to behave as follows:

$$\begin{aligned}
\boldsymbol{i}_s(t_{\mathrm{sw},1}) &= \boldsymbol{i}_s(0) + \boldsymbol{m}_0 t_{\mathrm{sw},1} \\
\boldsymbol{i}_s(t_{\mathrm{sw},2}) &= \boldsymbol{i}_s(t_{\mathrm{sw},1}) + \boldsymbol{m}_1 (t_{\mathrm{sw},2} - t_{\mathrm{sw},1}) \\
\boldsymbol{i}_s(t_{\mathrm{sw},3}) &= \boldsymbol{i}_s(t_{\mathrm{sw},2}) + \boldsymbol{m}_2 (t_{\mathrm{sw},3} - t_{\mathrm{sw},2}) \\
\boldsymbol{i}_s(T_s) &= \boldsymbol{i}_s(t_{\mathrm{sw},3}) + \boldsymbol{m}_3 (T_s - t_{\mathrm{sw},3}).
\end{aligned} \tag{3.13}$$

The approximated stator current evolution, the constant reference, the slopes and the three switching times are visualized in Fig. 3.2.

As was mentioned earlier, finding $\boldsymbol{t}_{\mathrm{sw}}$ is formulated as an optimization problem. The goal of the optimization problem is to find the three switching times that minimize the squared stator current error at times $\boldsymbol{t}_{\mathrm{sw}}$ and at T_s. The cost function for this

optimization problem is

$$J(k) = \sum_{\xi \in S} (\| \boldsymbol{i}_s^* - \boldsymbol{i}_s(k+\xi|k) \|_2^2), \tag{3.14}$$

where $S = \{\frac{t_{\mathrm{sw},1}}{T_s} \ \frac{t_{\mathrm{sw},2}}{T_s} \ \frac{t_{\mathrm{sw},3}}{T_s} \ 1\}$ and $\boldsymbol{i}_s$ is the predicted stator current. The prediction is done according to (3.13).

(3.13) and (3.14) can be combined, which results in

$$J(k) = \left\| \begin{bmatrix} \boldsymbol{i}_s^* \\ \boldsymbol{i}_s^* \\ \boldsymbol{i}_s^* \\ \boldsymbol{i}_s^* \end{bmatrix} - \begin{bmatrix} \boldsymbol{m}_0 & \mathbf{0} & \mathbf{0} \\ \boldsymbol{m}_0 - \boldsymbol{m}_1 & \boldsymbol{m}_1 & \mathbf{0} \\ \boldsymbol{m}_0 - \boldsymbol{m}_1 & \boldsymbol{m}_1 - \boldsymbol{m}_2 & \boldsymbol{m}_2 \\ \boldsymbol{m}_0 - \boldsymbol{m}_1 & \boldsymbol{m}_1 - \boldsymbol{m}_2 & \boldsymbol{m}_2 - \boldsymbol{m}_3 \end{bmatrix} \begin{bmatrix} t_{\mathrm{sw},1} \\ t_{\mathrm{sw},2} \\ t_{\mathrm{sw},3} \end{bmatrix} - \begin{bmatrix} \boldsymbol{i}_s(0) \\ \boldsymbol{i}_s(0) \\ \boldsymbol{i}_s(0) \\ \boldsymbol{i}_s(0) \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{m}_3 T_s \end{bmatrix} \right\|_2^2, \tag{3.15}$$

where $\mathbf{0}$ is the zero vector.

The complete optimization problem is

$$\begin{aligned} \underset{\boldsymbol{t}_{\mathrm{sw}}}{\text{minimize}} \quad & J(k) \\ \text{subject to} \quad & (2.37)\ (3.8) \text{ and } (3.9). \end{aligned} \tag{3.16}$$

This optimization problem is solved for each of the six previously mentioned switching orders. Doing so yields a vector of optimal times to switch for each of the six cases.

3.2.4 Part 3: Solving the Optimization Problem

As shown in Section 3.2.1, each of the six possible switching orders is associated with four different switch positions. To conveniently refer to all four of those vectors in the order they are to be applied in, a sequence of input vectors $\boldsymbol{U}_s$ is defined. Let the first of the four input vectors be $\boldsymbol{u}_0$, the second be $\boldsymbol{u}_1$, the third be $\boldsymbol{u}_2$ and the fourth be $\boldsymbol{u}_3$. Then, the sequence of input vectors is defined as

$$\boldsymbol{U}_s = \begin{bmatrix} \boldsymbol{u}_0 & \boldsymbol{u}_1 & \boldsymbol{u}_2 & \boldsymbol{u}_3 \end{bmatrix}, \tag{3.17}$$

where $\boldsymbol{U}_s \in \mathbb{R}^{3\times 4}$. Additionally, a set containing all six of the possible $\boldsymbol{U}_s$ is defined as $\mathcal{U}_s$.

Using these definitions, the goal of part three is to find the most suitable pair

$\{\boldsymbol{U}_s, \boldsymbol{t}_{\text{sw}}\}$ out of six such pairs. This is formulated as an optimization problem. To this end, the pair $\{\boldsymbol{U}_s, \boldsymbol{t}_{\text{sw}}\}$ that produced the lowest $J(k)$ in the previous step is selected.

The pseudo-code for direct MPC with fixed switching frequency with $N_P = 1$ is given in Algorithm 3.

Algorithm 3 Direct MPC with fixed switching frequency, Enumeration, $N_P = 1$

$\text{cost}_{\text{min}} \leftarrow \infty$
get $\boldsymbol{m}_0$
for $\boldsymbol{U}_s \in \mathcal{U}_s$ **do**
  get $\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3$
  get $\boldsymbol{t}_{\text{sw}}$ by solving (3.16)
  $\text{cost} \leftarrow J(k)$
  if $\text{cost} < \text{cost}_{\text{min}}$ **then**
    $\text{cost}_{\text{min}} \leftarrow \text{cost}$
    $\{\boldsymbol{U}_{s,\text{opt}}, \boldsymbol{t}_{\text{sw,opt}}\} \leftarrow \{\boldsymbol{U}_s, \boldsymbol{t}_{\text{sw}}\}$
  end if
end for
apply $\boldsymbol{U}_{s,\text{opt}}$ at times $\boldsymbol{t}_{\text{sw,opt}}$
repeat

4. SIMULATION RESULTS AND ANALYSIS

A total of four different control strategies have been introduced in previous chapters: FOC/CB-PWM, FCS-MPC, VSP^{2}CC and direct MPC with fixed switching frequency. The performance of all four is studied by simulating them under different conditions using Matlab.

In all simulations, the IM model given in Chapter 2 is controlled by an ideal two-level inverter with a constant dc-link voltage. The speed and torque controllers are omitted, meaning that each control strategy is given a stator current reference to follow.

This chapter has three sections. In Section 4.1, the performance of the four strategies is analyzed during nominal steady-state operation. In Section 4.2, the dynamical performance of the system, under torque reference changes, is examined. Finally, in Section 4.3, the control strategies are compared in terms of total harmonic distortion (THD) under different switching frequencies.

4.1 Steady-state Performance

In this section, the stator current reference $\boldsymbol{i}^*_{s,dq}$ is set to a value that produces the nominal torque and the desired degree of magnetization. The stator current reference is kept constant to study steady-state behavior of the four different control strategies. The stator currents, switch positions, stator flux magnitude, torque and stator current harmonics are displayed for each of the four control strategies.

The steady-state performance is studied using nominal values and an average switching frequency of 1050Hz. For asymmetric regularly sampled CB-PWM this means a carrier frequency to fundamental frequency ratio $\frac{f_c}{f_f}$ of 21. For direct MPC with fixed switching frequency, it means a sampling time of $\frac{1}{2 \cdot 1050\text{Hz}} \approx 476\,\mu\text{s}$. Both of these produce a constant switching frequency of exactly 1050Hz.

FCS-MPC and VSP^{2}CC are simulated with a sampling interval of 25 µs. They both use λ to balance the importance of current THD in relation to the switching effort.

Because of this, it is not possible to achieve an average switching frequency of exactly 1050Hz. For FCS-MPC, $\lambda = 0.0022475$ produces an average switching frequency of 1050.5Hz and for VSP^{2}CC, $\lambda = 0.004398$ produces 1050.4Hz. These values for λ are used.

4.1.1 Carrier-based Pulse Width Modulation

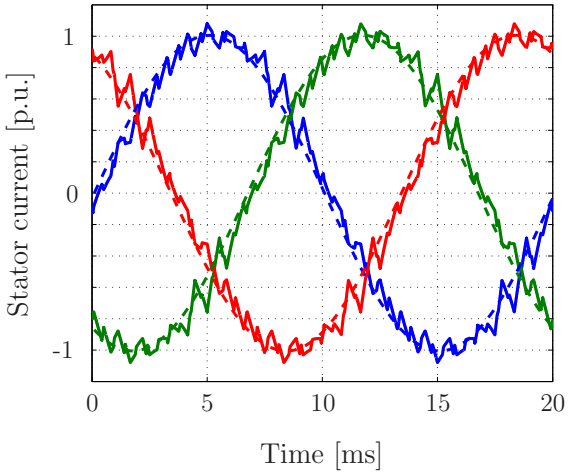

Figure 4.1 *CB-PWM steady-state stator currents*

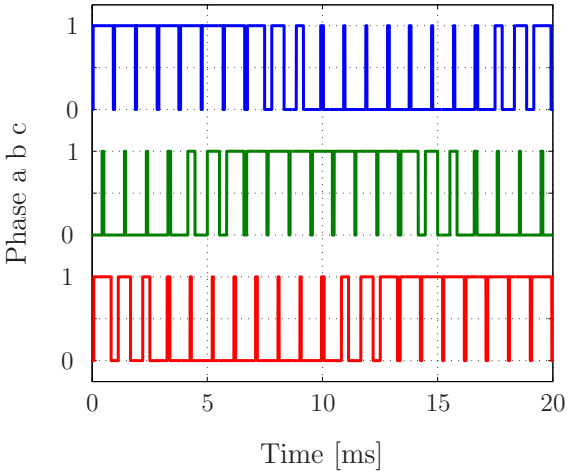

Figure 4.2 *CB-PWM steady-state switch position*

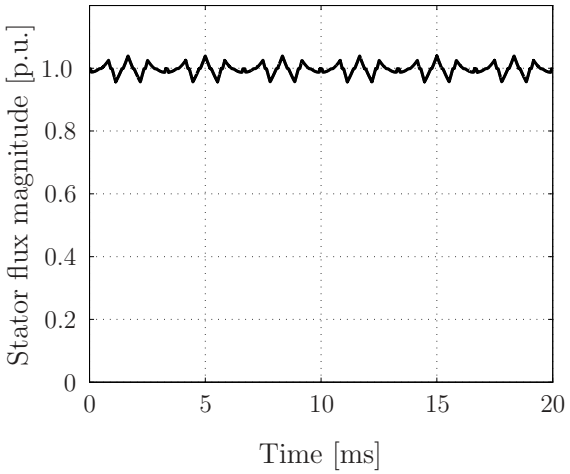

Figure 4.3 *CB-PWM steady-state stator flux magnitude*

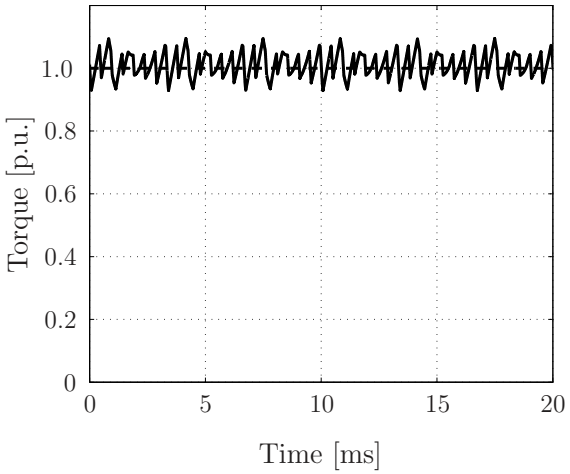

Figure 4.4 *CB-PWM steady-state electromagnetic torque*

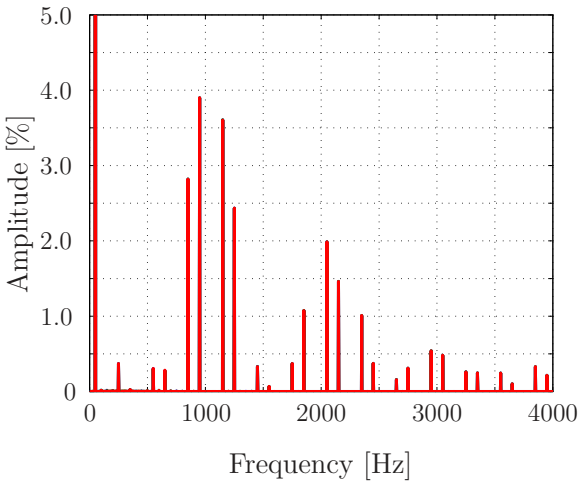

Figure 4.5 *CB-PWM stator current harmonics*

Asymmetric regularly sampled CB-PWM is used with the min/max -based common-mode injection, as explained in Section 2.4.1. This provides quarter-wave symmetry for the switch positions and odd, non-triplen harmonics for the current.

As can be seen from the above figures, the stator current references and the torque reference are being tracked. The stator flux magnitude is 1 p.u, as expected. Quarter-wave symmetry can be observed in the switch positions.

The five more prominent harmonics are at 850Hz, 950Hz, 1150Hz, 1250Hz and 2050Hz, which are the 17th, 19th, 23rd, 25th and 41st. These are odd, non-triplen integer multiples of the fundamental frequency.

4.1.2 Finite Control Set Model Predictive Control

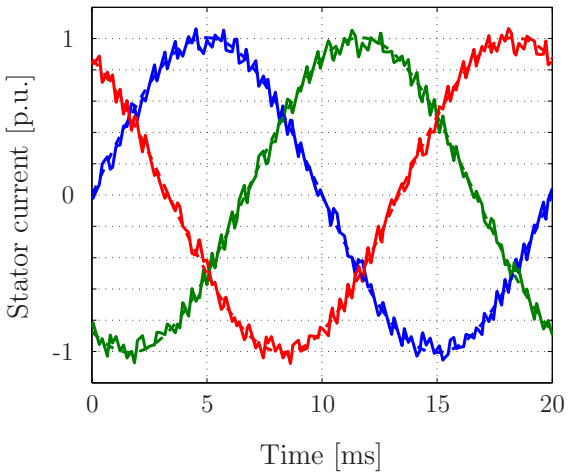

Figure 4.6 *FCS-MPC steady-state stator currents*

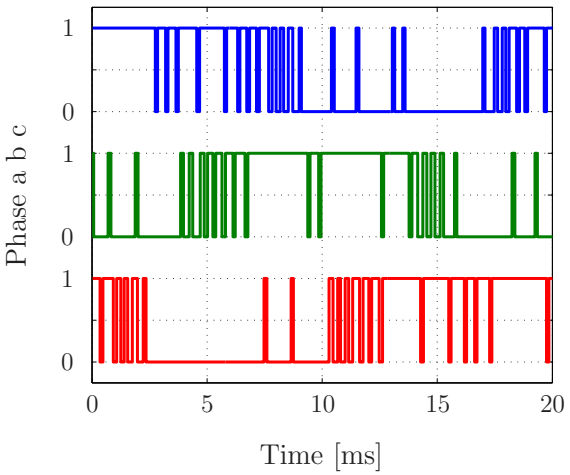

Figure 4.7 *FCS-MPC steady-state switch position*

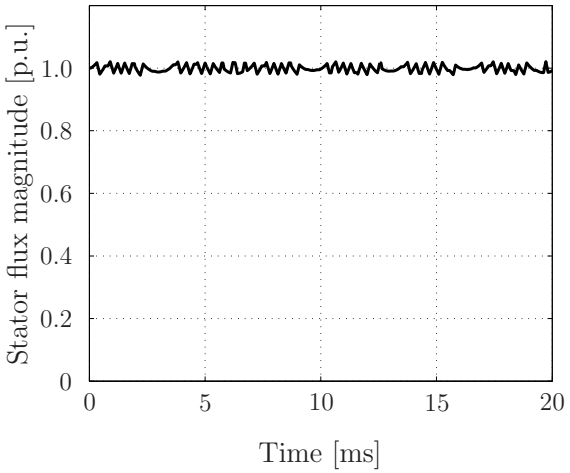

Figure 4.8 *FCS-MPC steady-state stator flux magnitude*

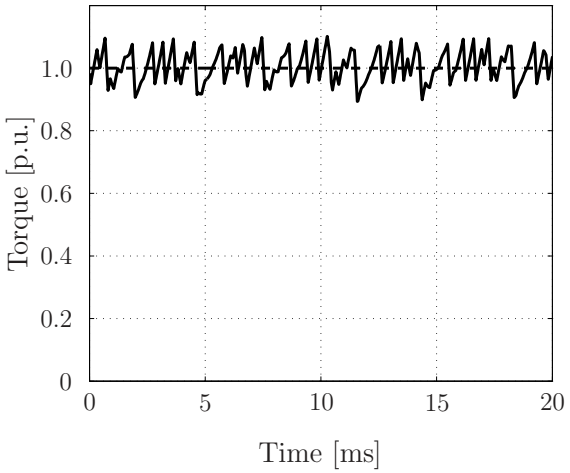

Figure 4.9 *FCS-MPC steady-state electromagnetic torque*

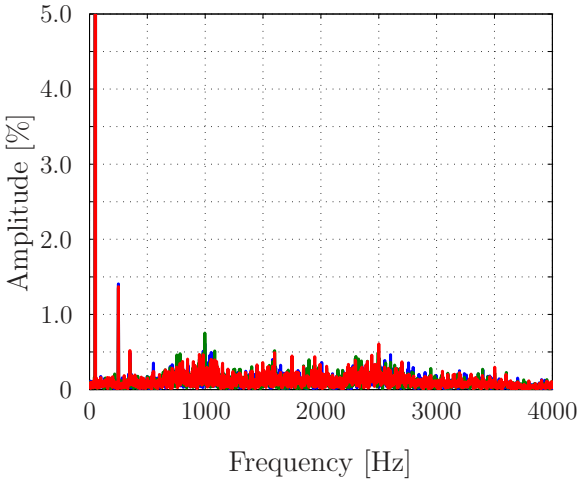

Figure 4.10 *FCS-MPC stator current harmonics*

In FCS-MPC, switching occurs when the stator current error penalty would be too high without switching. As a consequence, an inverter leg may switch back and forth repeatedly between sampling intervals or it might not switch at all for a long time. In other words, the need to switch is evaluated for each sampling interval on a case by case basis without any concern for symmetries. This explains the lack of symmetry in all of the figures, which, in turn, explains why the stator current harmonics are spread over many frequencies.

The stator current references are tracked well, the stator flux magnitude is 1 p.u. and the the torque reference is tracked. In other words, the control works.

4.1.3 Variable Switching Point Predictive Current Control

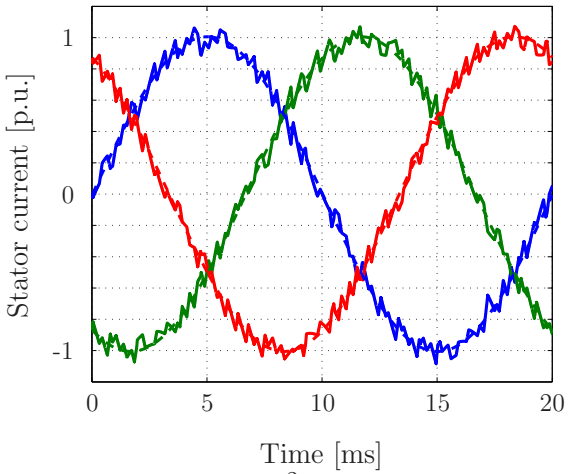

Figure 4.11 *VSP^{2}CC steady-state stator currents*

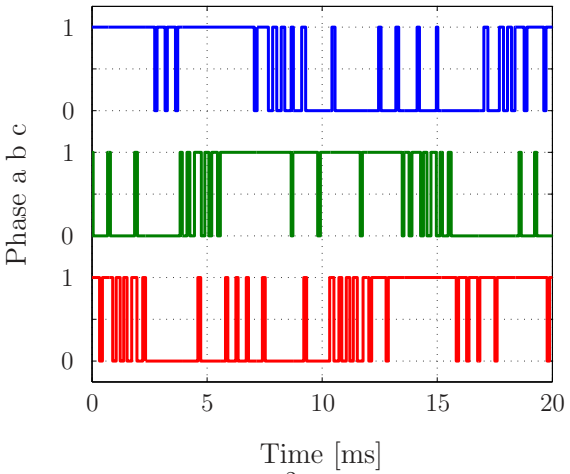

Figure 4.12 *VSP^{2}CC steady-state switch position*

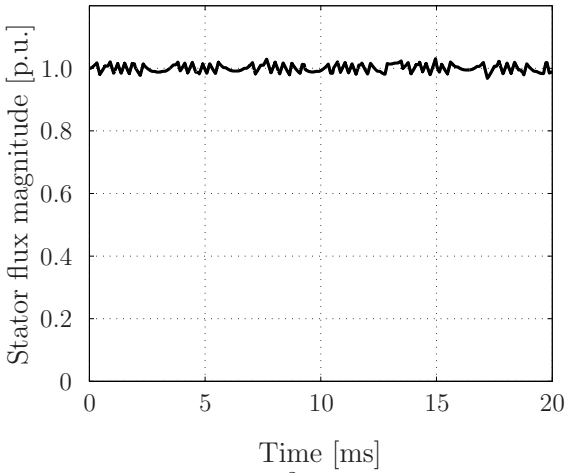

Figure 4.13 *VSP^{2}CC steady-state stator flux magnitude*

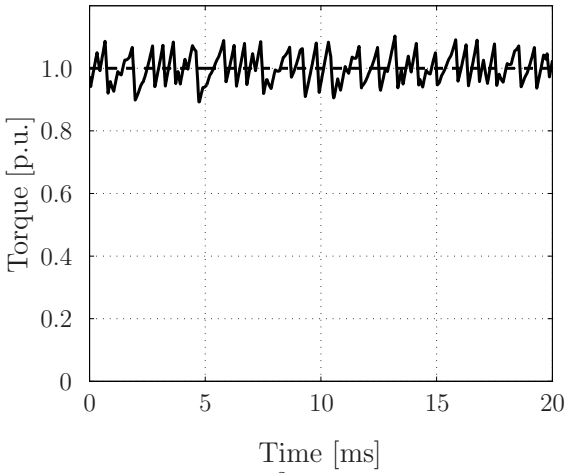

Figure 4.14 *VSP^{2}CC steady-state electromagnetic torque*

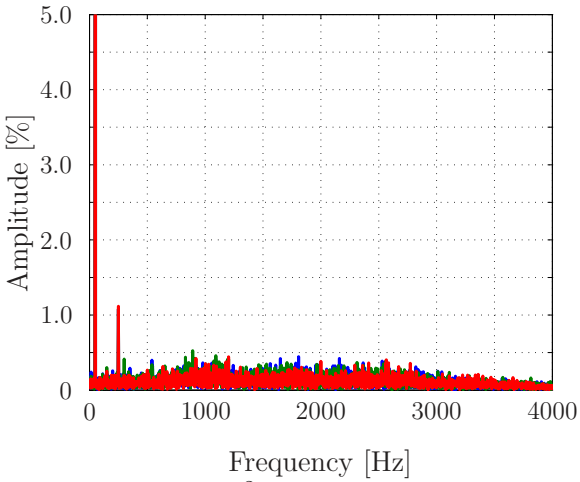

Figure 4.15 *VSP²CC stator current harmonics*

As a part of VSP²CC, an optimal time to switch is calculated. Consequently, the time to switch can be delayed within a sampling interval. The frequency components of the stator currents are spread over many frequencies due to the significant lack of symmetries in the switch positions, and consequently in the produced currents.

Both the stator current references and the torque reference are tracked. The stator flux magnitude stays very close to 1 p.u.

4.1.4 Direct MPC with Fixed Switching Frequency

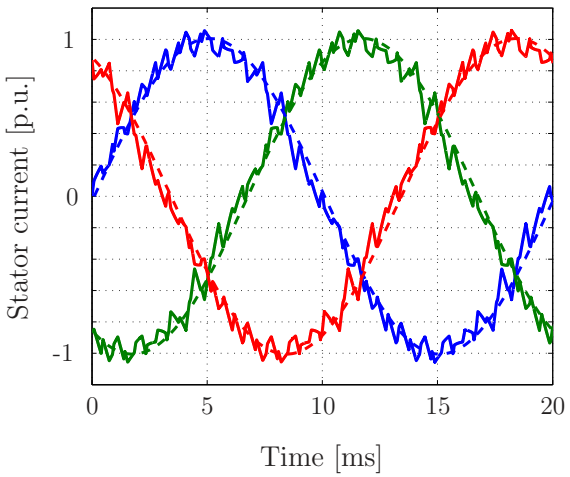

Figure 4.16 *Direct MPC with fixed switching frequency, steady-state stator currents*

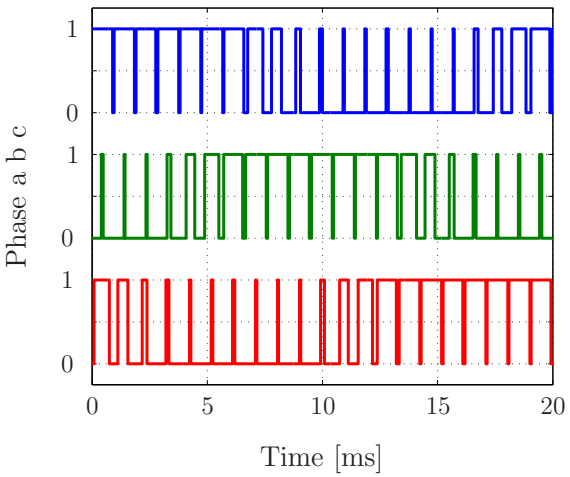

Figure 4.17 *Direct MPC with fixed switching frequency, steady-state switch position*

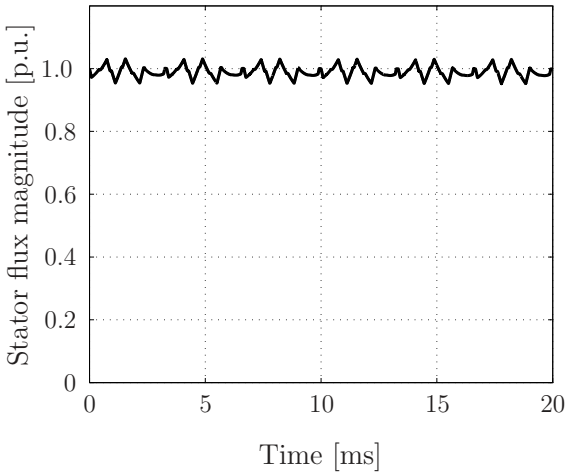

Figure 4.18 *Direct MPC with fixed switching frequency, steady-state stator flux magnitude*

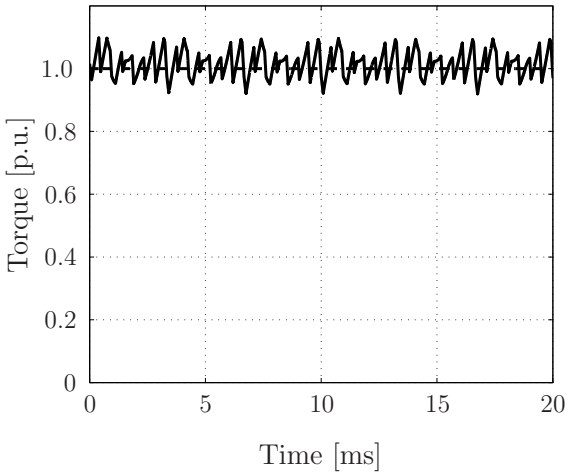

Figure 4.19 *Direct MPC with fixed switching frequency, steady-state electromagnetic torque*

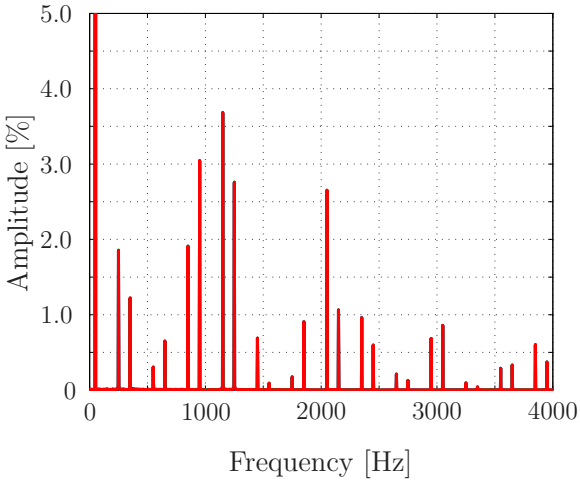

Figure 4.20 *Direct MPC with fixed switching frequency, stator current harmonics*

In direct MPC with fixed switching frequency the switch positions resemble those of CB-PWM. The inverter legs are set to switch once in each sampling interval. CB-PWM does the same within the linear modulation range. Unlike CB-PWM, direct MPC with fixed switching frequency dose not impose any symmetries. However, the harmonic spectrum in Fig. 4.20 implies that sequences of switch positions are repeated symmetrically.

In addition to the switch positions, the stator current harmonics of direct MPC with fixed switching frequency are somewhat similar to those of CB-PWM. The five more significant harmonics are at the same frequencies as for CB-PWM, although their magnitudes differ. The locations of the harmonics seem to follow those of CB-PWM, to an extent. One clear difference is that with direct MPC with fixed

switching frequency, the fifth, seventh and eleventh harmonics are noticeably larger. In CB-PWM, those harmonics appear in overmodulation.

The stator current references are tracked successfully. The stator flux magnitude stays close to the desired 1 p.u. Additionally, the torque reference is tracked. In other words, direct MPC with fixed switching frequency manages to track all of its reference values in steady-state operation.

4.2 Torque Step Response

In this section, the torque transient performance of each of the four control strategies is displayed and analyzed. While operating at the steady-state operating conditions stated in Section 4.1., the torque reference is stepped down from 1 to 0 p.u. at t = 5ms. The current reference is changed accordingly. After 10ms, i.e. at 15ms, the torque reference, and consequently the current reference are set equal to their nominal values. The stator currents and their references, the switch positions, the stator flux magnitude and the torque are displayed.

In CB-PWM and direct MPC with fixed switching frequency, average switching frequency f_{sw} is a function of sampling interval length T_s. Consequently, to achieve a particular f_{sw}, T_s needs to be set accordingly. Additionally, any discrete control algorithm will react to a reference change only when the next sampling instance occurs. Therefore, any reference changes occurring in between of sampling instances will be reacted to only after a slight delay. For these reasons, a slight delay in the responses can be seen with CB-PWM and direct MPC with fixed switching frequency. The sampling intervals for FCS-MPC and VSP^{2}CC happen to be such that the transient occurs as the sampling is occurring.

The same sampling intervals and λ are used as in the previous section. The targeted average switching frequency is 1050Hz.

4.2.1 Carrier-based Pulse Width Modulation

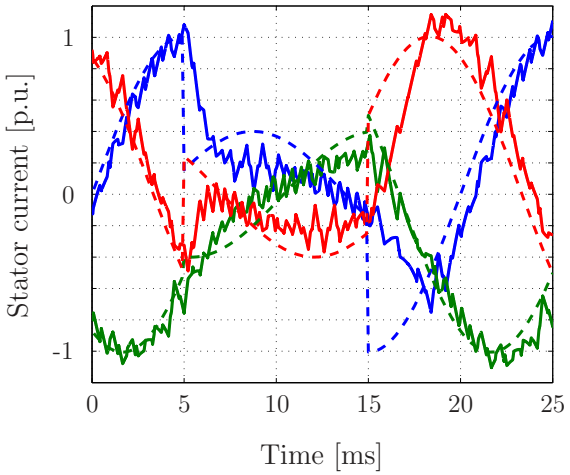

Figure 4.21 *CB-PWM stator currents*

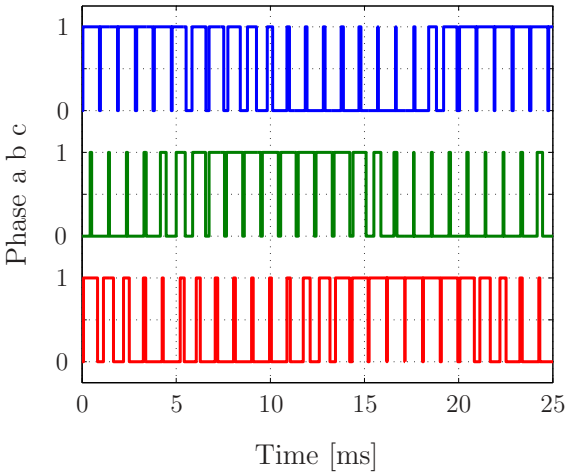

Figure 4.22 *CB-PWM switch position*

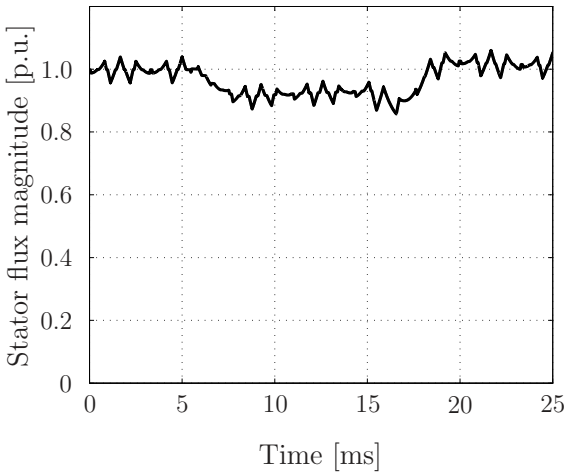

Figure 4.23 *CB-PWM stator flux magnitude*

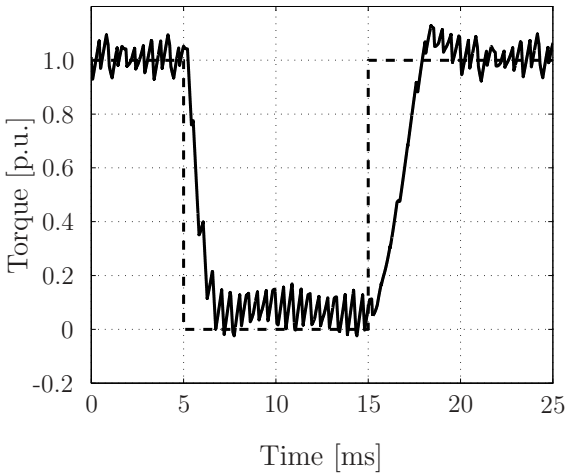

Figure 4.24 *CB-PWM electromagnetic torque*

CB-PWM requires a three-phase voltage reference, which is created from the current references by using PI controllers. In a step change, the controller output changes proportional to the error signal. Over time, the integral part of the PI controller removes the steady-state error that the proportional part alone would produce.

In Fig. 4.24, after the first step at 5ms, the torque starts to approach zero. As the error gets smaller, the rate of change of the torque slows down. Eventually, the integral part removes the error over time. At the second step change, the torque has a small overshoot.

In CB-PWM, the inverter legs are set to switch once within a sampling interval. This

causes the torque and currents to temporarily divert from their references during the step changes.

4.2.2 Finite Control Set Model Predictive Control

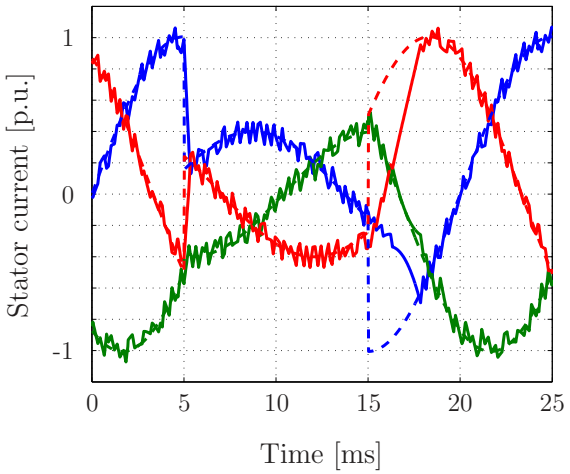

Figure 4.25 *FCS-MPC stator currents*

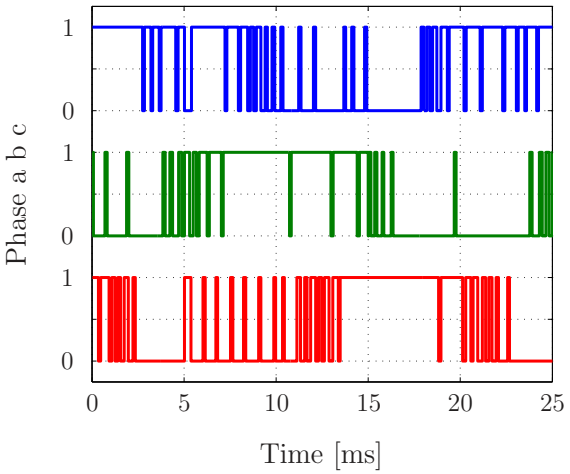

Figure 4.26 *FCS-MPC switch position*

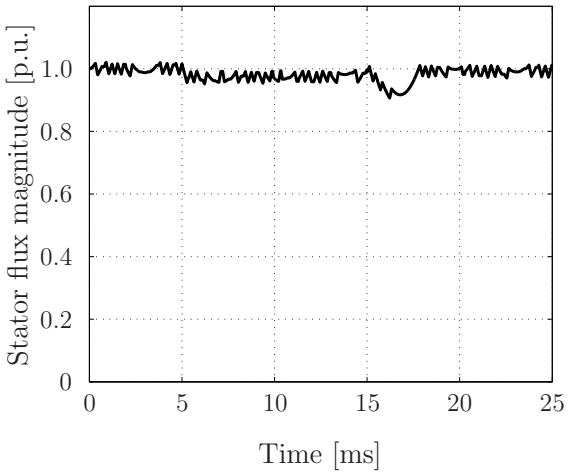

Figure 4.27 *FCS-MPC stator flux magnitude*

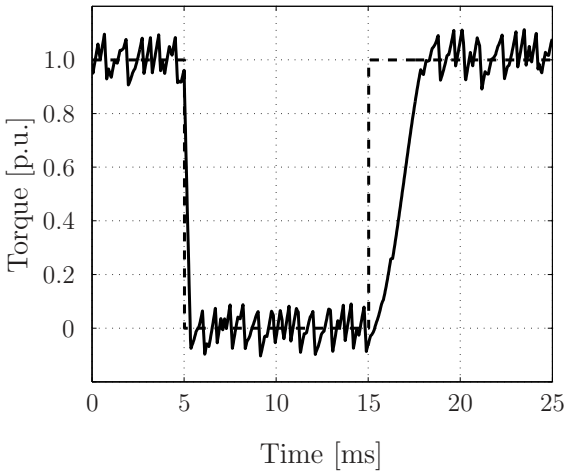

Figure 4.28 *FCS-MPC electromagnetic torque*

FCS-MPC has the freedom to apply any switch position for a duration of one sampling interval. This means that it can produce a torque step response that is very close to the theoretical best possible response. Such behavior can be seen in the current and torque responses above.

At 5ms, the inverter legs corresponding to phases a and c switch positions and stay at those positions for as long as is necessary. This inversion of the voltages allows

for a very fast response time. At 15ms, the response is a lot slower. Because the IM operates at nominal speed, thus the settling time is only limited by the available dc-link voltage [8].

4.2.3 Variable Switching Point Predictive Current Control

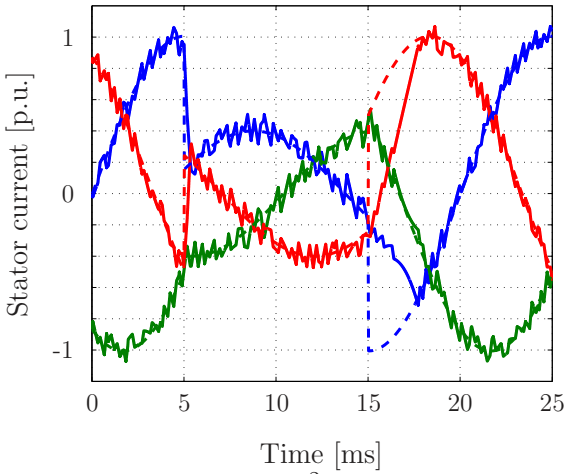

Figure 4.29 *VSP²CC stator currents*

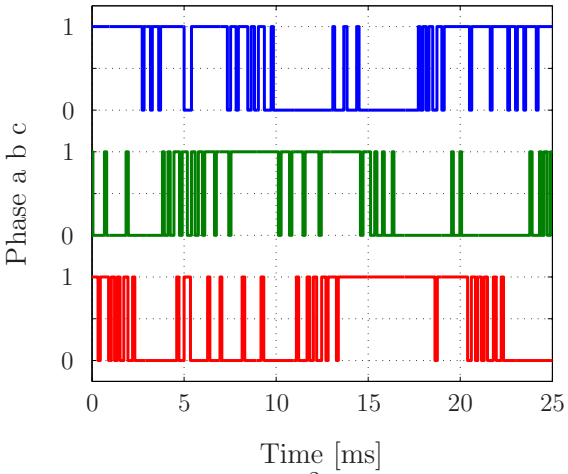

Figure 4.30 *VSP²CC switch position*

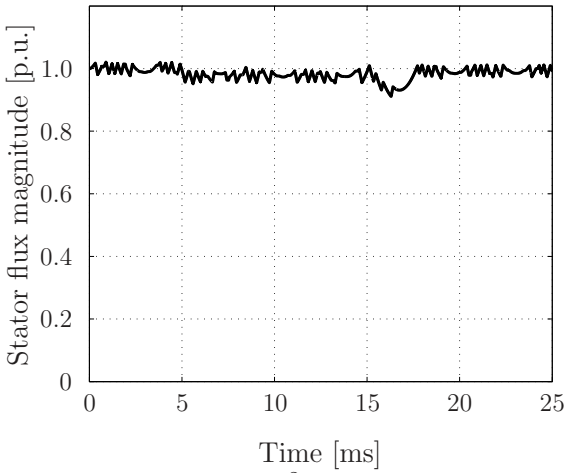

Figure 4.31 *VSP²CC stator flux magnitude*

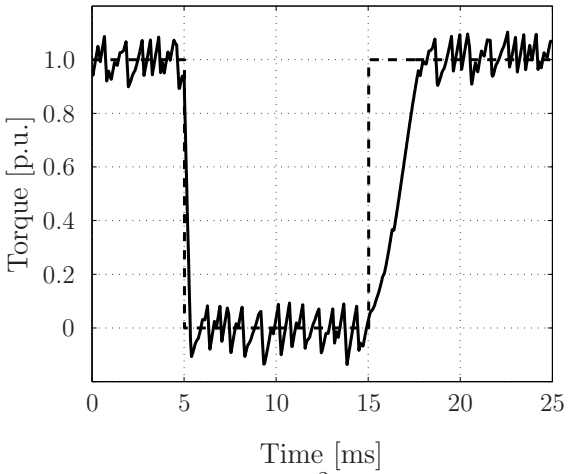

Figure 4.32 *VSP²CC electromagnetic torque*

VSP²CC has similar degrees of freedom as FCS-MPC when it comes to switching. Consequently, the torque response is as fast as with FCS-MPC. Even though the currents and the torque are very similar to those of FCS-MPC, the switch positions are different. The current and torque responses are fast and there is no overshoot once the references are reached.

Similarly to FCS-MPC, the inversion of voltages occurs at 5ms. The response is slower at 15ms than at 5ms. This can be explained by the same limitations regarding the available dc-link voltage as in FCS-MPC.

4.2.4 Direct MPC with Fixed Switching Frequency

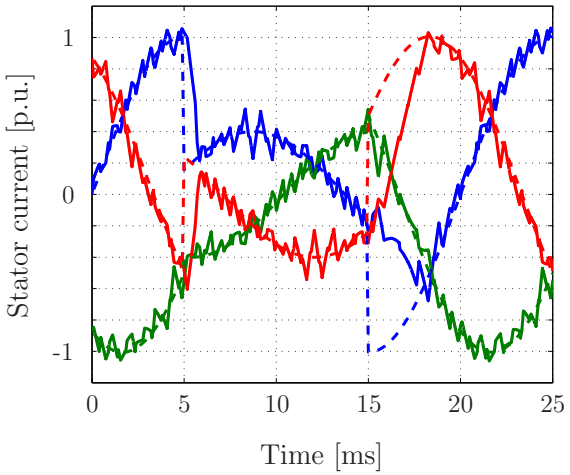

Figure 4.33 *Direct MPC with fixed switching frequency, stator currents*

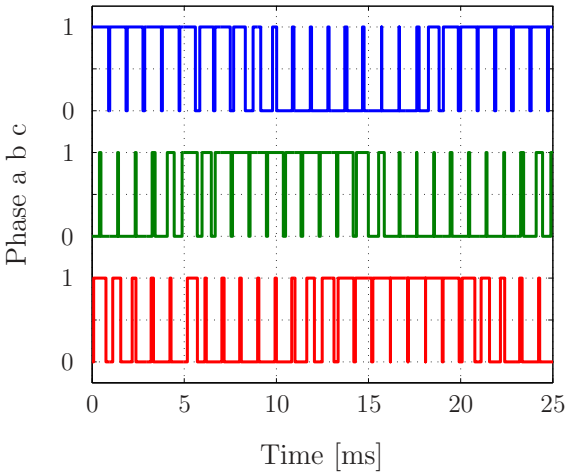

Figure 4.34 *Direct MPC with fixed switching frequency, switch position*

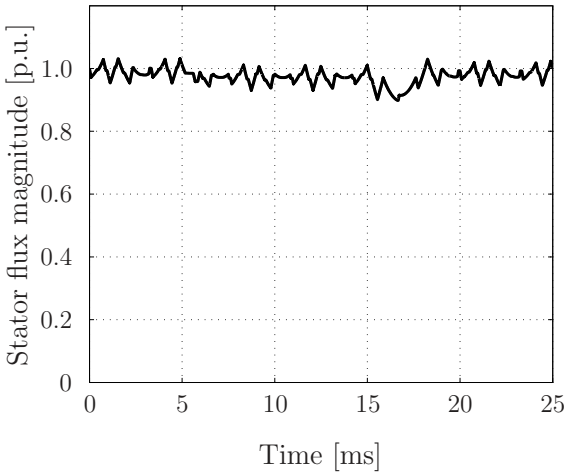

Figure 4.35 *Direct MPC with fixed switching frequency, stator flux magnitude*

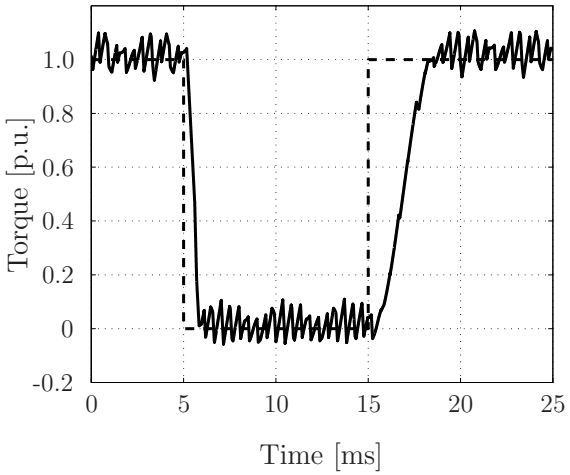

Figure 4.36 *Direct MPC with fixed switching frequency, electromagnetic torque*

As mentioned earlier, direct MPC with fixed switching frequency relies on selecting an appropriate sampling interval for setting the switching frequency. As a consequence, the information about the reference changing has not reached the algorithm at exactly 5ms, but slightly after. Considering this delay, the step response is very close to that of FCS-MPC and VSP^{2}CC.

The difference between FCS-MPC/VSP^{2}CC responses and the direct MPC with fixed switching frequency response comes from the constraints in direct MPC with fixed switching frequency. It has to have all of the inverter legs switch once within a sampling interval, even when it is not optimal to do so. However, unlike in CB-PWM, the undesired leg position is kept for only the minimum time possible.

4.3 THD Comparison

In this section, the four different control strategies, CB-PWM, FCS-MPC, VSP^{2}CC and direct MPC with fixed switching frequency, are compared in terms of stator current THD. More specifically, steady-state simulations over a wide range of switching frequencies, i.e., from 550Hz to 1650Hz, are run and the stator current THD is recorded. Specifically, for each simulation, the average switching frequency and the stator current THD are recorded.

With CB-PWM, to half the THD, the switching frequency needs to double, and vice versa. As a consequence, the relation between THD and switching frequency can be expressed as

$$\mathrm{THD}(\boldsymbol{i}_s) f_{\mathrm{sw}} = c, \tag{4.1}$$

where c is a constant [8]. This product c can be used as a helpful metric when comparing the performance of different control strategies. A lower c means that with the same switching frequency, a lower THD is achieved, and vice versa. With MPC algorithms, the product expressed with c is not necessarily a constant as it is with CB-PWM. This necessitates plotting c as a function of the average switching frequency as a method to compare the performance of control schemes.

Stator current THD as a function of average switching frequency for the four different control strategies is given in Fig. 4.37. The product c as a function of the average switching frequency is shown in Fig. 4.38. In both figures, the CB-PWM performance is shown in black, direct MPC with fixed switching frequency in red, VSP^{2}CC in blue and FCS-MPC in green.

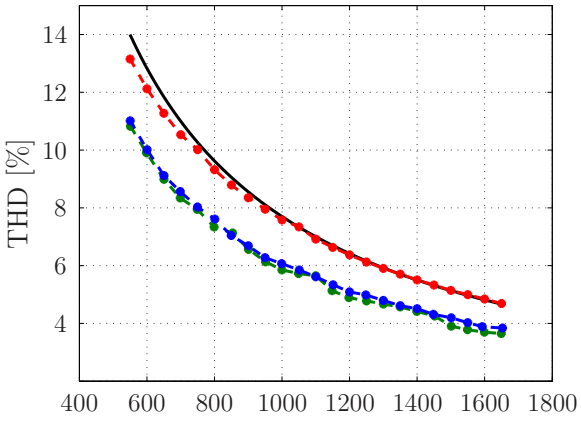

Figure 4.37 *THD comparison*

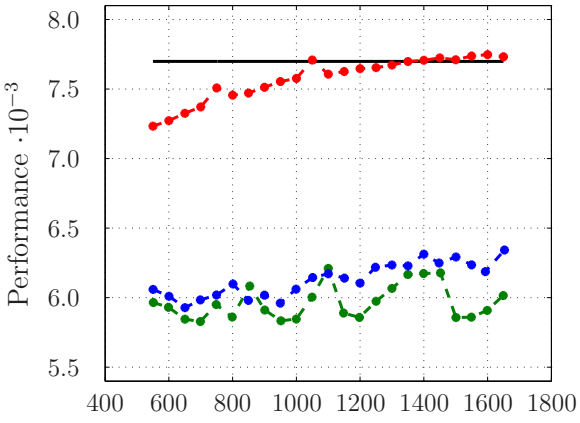

Figure 4.38 *Product c comparison*

In Fig. 4.37, with lower switching frequencies, the THD is high regardless of the control algorithm. The simulated machine is a medium voltage (MV) IM, which are sometimes used in conjunction with multilevel inverters to deal with this issue. Also, if a low voltage (LV) IM was used instead, a higher switching frequency would be acceptable. The comparison is still valuable, since even if the machine parameters change, the results stay qualitatively similar.

As was mentioned, the performance metric c is constant for CB-PWM. With direct MPC with fixed switching frequency, c is almost the same as for CB-PWM, only slightly lower. The other MPC based algorithms have a much lower c. The value for c fluctuates, but the trend is that a lower switching frequency provides a better c.

5. CONCLUSIONS

MPC is a control algorithm in which a model, an optimization problem and receding horizon policy are used when deciding on which control actions to take. The model is used for predicting the states and outputs of the controlled system as a function of inputs and previous states. The optimization problem consists of the objective function and constraints. The objective function maps the control objectives into a scalar and the constraints restrict states, inputs and outputs with equalities and inequalities. Solving the optimization problem yields the sequence of inputs that, according to the model, result in the most desirable system behavior. Closed loop control of the system is introduced by using receding horizon policy, i.e., by repeatedly solving the optimization problem after the first optimal input is applied.

MPC can be used to control MIMO or SISO systems and linear or non-linear systems. In power electronics, the controlled systems are often non-linear can have multiple inputs or outputs. For example, a VSD is a MIMO system. Traditionally, this is addressed by averaging and cascaded control loops. With MPC, these can be avoided. Additionally, power electronic systems can be modeled fairly accurately. For these reasons, given that the computing power is continuously increasing, MPC for ac drives is a viable alternative to consider.

An MPC algorithm for a drive system consisting of a two-level inverter and an IM that can operate the power electronic converter at a fixed switching frequency was proposed. The algorithm, i.e., direct MPC with fixed switching frequency, controls the switches directly. This means that it does not require a modulator. With the proposed method, the optimization variable is a vector that consists of switching times. As a consequence, the optimization problem underlying MPC yields the best instants each inverter leg should switch, provided that they are switched in a particular order.

The proposed method contains a constraint forcing each inverter leg to switch once within a sampling interval, even if switching is not optimal. For a two-level inverter, the constraint means that the optimization problem is solved six times for each time step. This constraint causes a fixed switching frequency that is defined by the

chosen sampling interval. It also causes the frequency content of the stator currents to be at integer multiples of the fundamental frequency. These distinct harmonics are not present when using some of the existing direct MPC algorithms (FCS-MPC, VSP^{2}CC).

The proposed method achieves a steady-state performance on par with CB-PWM. The other MPC algorithms studied outperformed the proposed method in terms of steady-state THD. Under a torque step change, the proposed method outperforms CB-PWM. The step responses of direct MPC with fixed switching frequency are almost as good as with FCS-MPC and VSP^{2}CC.

If distinct harmonics and a fixed switching frequency are desired, the proposed method is a viable alternative to CB-PWM, as it performs better during torque transients. Additionally, the proposed method is good for MIMO systems. If the harmonics are allowed to be distributed along random frequencies, FCS-MPC or VSP^{2}CC should be used, as they perform the best in steady-state situations.

5.1 Future Research

During this thesis, all MPC algorithms had a prediction horizon length of one. As mentioned in Section 2.7, MPC algorithms benefit from having a longer prediction horizon. The proposed algorithm should be extended to allow a longer prediction horizon in a computationally efficient manner.

An MV IM can be controlled with a three-level neutral point clamped (NPC) inverter [19]. This allows for a lower average switching frequency and a better THD. A three-level inverter introduces three positions for an inverter leg: -1, 0 and 1. The proposed algorithm should be developed to work with a three-level inverter and its performance with the necessary changes is to be studied.

The thesis focuses only on the motor control part of an ac drive system. The suitability of the proposed method for grid-side control should be explored.

Direct MPC with fixed switching frequency should be implemented, for example on a field programmable gate array (FPGA). This would allow verification of the performance of the proposed algorithm with a hardware-in-the-loop simulation or a real drive system.

BIBLIOGRAPHY

[1] A. Ayad, P. Karamanakos, and R. Kennel, "Variable switching point predictive current control of quasi-Z-source inverters," in *2017 IEEE Applied Power Electronics Conference and Exposition (APEC)*, March 2017, pp. 2773–2780.

[2] F. Blaschke, "Principle of field orientation as used in the new transvektor control system for induction machines, (das prinzip der feldorientierung, die grundlage fuer die transvektor- regelung von drehfeldmaschinen)," *Siemens Z*, vol. 45, pp. 757–760, 1971.

[3] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[4] D. Brock and G. Moore, *Understanding Moore's Law: Four Decades of Innovation*. Chemical Heritage Foundation, 2006.

[5] P. Cortes, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo, and J. Rodriguez, "Predictive control in power electronics and drives," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 12, pp. 4312–4324, Dec 2008.

[6] W. C. Duesterhoeft, M. W. Schulz, and E. Clarke, "Determination of instantaneous currents and voltages by means of alpha, beta, and zero components," *Transactions of the American Institute of Electrical Engineers*, vol. 70, no. 2, pp. 1248–1255, July 1951.

[7] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, Apr. 1965.

[8] T. Geyer, *Model Predictive Control of High Power Converters and Industrial Drives*. Wiley, 2016.

[9] T. Geyer, P. Karamanakos, and R. Kennel, "On the benefit of long-horizon direct model predictive control for drives with *LC* filters," in *2014 IEEE Energy Conversion Congress and Exposition (ECCE)*, Sept 2014, pp. 3520–3527.

[10] T. Geyer and D. E. Quevedo, "Multistep finite control set model predictive control for power electronics," *IEEE Transactions on Power Electronics*, vol. 29, no. 12, pp. 6836–6846, Dec 2014.

[11] ——, "Performance of multistep finite control set model predictive control for power electronics," *IEEE Transactions on Power Electronics*, vol. 30, no. 3, pp. 1633–1644, March 2015.

[12] K. Hasse, "Zum dynamischen verhalten der asynchronmachine bei betriek mit variabler standerfrequenz und standerspannung," *ETZ-A*, vol. 89, pp. 387–391, 1968.

[13] D. Holmes and T. Lipo, *Pulse Width Modulation for Power Converters: Principles and Practice*, ser. IEEE Press Series on Power Engineering. John Wiley & Sons, 2003.

[14] P. Karamanakos, T. Geyer, and R. Kennel, "On the choice of norm in finite control set model predictive control," *IEEE Transactions on Power Electronics*, vol. PP, no. 99, pp. 1–1, 2017.

[15] P. Karamanakos, T. Geyer, T. Mouton, and R. Kennel, "Computationally efficient sphere decoding for long-horizon direct model predictive control," in *2016 IEEE Energy Conversion Congress and Exposition (ECCE)*, Sept 2016, pp. 1–8.

[16] P. Karamanakos, T. Geyer, N. Oikonomou, F. D. Kieferndorf, and S. Manias, "Direct model predictive control: A review of strategies that achieve long prediction intervals for power electronics," *IEEE Industrial Electronics Magazine*, vol. 8, no. 1, pp. 32–43, March 2014.

[17] P. Karamanakos, "Model predictive control strategies for power electronics converters and ac drives," Ph.D. dissertation, National Technical University of Athens, July 2013.

[18] P. Krause, O. Wasynczuk, S. Sudhoff, and I. P. E. Society, *Analysis of electric machinery and drive systems*, ser. IEEE Press series on power engineering. IEEE Press, 2002.

[19] A. Nabae, I. Takahashi, and H. Akagi, "A new neutral-point-clamped PWM inverter," *IEEE Transactions on Industry Applications*, vol. IA-17, no. 5, pp. 518–523, Sept 1981.

[20] Y. Nesterov and A. Nemirovskii, *Interior-point Polynomial Algorithms in Convex Programming*, ser. Studies in Applied Mathematics. Society for Industrial and Applied Mathematics, 1994.

[21] R. H. Park, "Two-reaction theory of synchronous machines generalized method of analysis-part I," *Transactions of the American Institute of Electrical Engineers*, vol. 48, no. 3, pp. 716–727, July 1929.

[22] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733 – 764, 2003.

[23] D. E. Quevedo, G. C. Goodwin, and J. A. De Don, "Finite constraint set receding horizon quadratic control," *International Journal of Robust and Nonlinear Control*, vol. 14, no. 4, pp. 355–377, 2004.

[24] P. Stolze, P. Karamanakos, R. Kennel, S. Manias, and C. Endisch, "Effective variable switching point predictive current control for ac low-voltage drives," *International Journal of Control*, vol. 88, no. 7, pp. 1366–1378, 2015.

[25] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, "Model predictive control for power converters and drives: Advances and trends," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 935–947, Feb 2017.

[26] L. Wolsey, *Integer Programming*, ser. Wiley Series in Discrete Mathematics and Optimization. Wiley, 1998.

APPENDIX A. PER UNIT SYSTEM

The goal in using per unit values is to normalize the system under analysis, which is done by using base values. To transform actual values to per unit values, the actual values are divided by the corresponding base values. To transform per unit values to SI units, the per unit values are multiplied by the base value of that unit.

The base values are chosen according to nominal values of the IM used in the thesis. The base values for voltage, current and angular speed are chosen, whereas the rest are derived from them. The base voltage V_B is the peak value of the nominal phase voltage of the machine. The base current I_B is the peak value of the nominal stator current and the base angular speed ω_B is 2π times the nominal stator frequency.

The base values used are given in Table 1. pf stands for power factor.

Table 1 *The base values used*

Base value of	**Symbol**	**Base value**
Voltage	V_B	$\sqrt{\frac{2}{3}}V_R$
Current	I_B	$\sqrt{2}I_R$
Angular speed	ω_B	$\omega_{s,R}$
Frequency	f_B	$\frac{\omega_B}{2\pi}$
Impedance	Z_B	$\frac{V_B}{I_B}$
Inductive reactance	$X_{L,B}$	$\frac{Z_B}{\omega_B}$
Capacitive reactance	$X_{C,B}$	$\frac{1}{\omega_B Z_B}$
Apparent power	S_B	$\frac{3}{2}V_B I_B$
Torque	T_B	$\frac{\text{pf } pS_B}{\omega_B}$